

A fully-nonlocal energy-based formulation and high-performance realization of the quasicontinuum method

Thesis by
Jeff Amelang

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy



California Institute of Technology
Pasadena, California

2016
(Defended June 24, 2015)

© 2016

Jeff Amelang

All Rights Reserved

Acknowledgments

I gratefully acknowledge support from the Department of Energy National Nuclear Security Administration (NNSA) under award number DE-FC52-08NA28613 as well as from the National Science Foundation (NSF) under grant number CMMI-123436.

I'm grateful to Michael for our many hours of discussions in his office as well as his guidance and encouragement during my first years at Caltech. Much of what I learned about plasticity was in Ravi's class, and his sincere concern for the students and for me was appreciated. Mathieu gave me a chance at teaching a class at Caltech, a unique opportunity which became foundational for my career.

Of everyone at Caltech, I'm thankful for the invaluable patience, mentorship, and friendship that Dennis offered me over these years. His whiteboard may never recover.

Most of all, I thank my wife and best friend Aubrie for her support, inspiration, and inexhaustible kindness. You are the world to me.

Abstract

The quasicontinuum (QC) method was introduced to coarse-grain crystalline atomic ensembles in order to bridge the scales from individual atoms to the micro- and mesoscales. Though many QC formulations have been proposed with varying characteristics and capabilities, a crucial cornerstone of all QC techniques is the concept of summation rules, which attempt to efficiently approximate the total Hamiltonian of a crystalline atomic ensemble by a weighted sum over a small subset of atoms. In this work we propose a novel, fully-nonlocal, energy-based formulation of the QC method with support for legacy and new summation rules through a general energy-sampling scheme. Our formulation does not conceptually differentiate between atomistic and coarse-grained regions and thus allows for seamless bridging without domain-coupling interfaces. Within this structure, we introduce a new class of summation rules which leverage the affine kinematics of this QC formulation to most accurately integrate thermodynamic quantities of interest. By comparing this new class of summation rules to commonly-employed rules through analysis of energy and spurious force errors, we find that the new rules produce no residual or spurious force artifacts in the large-element limit under arbitrary affine deformation, while allowing us to seamlessly bridge to full atomistics. We verify that the new summation rules exhibit significantly smaller force artifacts and energy approximation errors than all comparable previous summation rules through a comprehensive suite of examples with spatially non-uniform QC discretizations in two and three dimensions. Due to the unique structure of these summation rules, we also use the new formulation to study scenarios with large regions of free surface, a class of problems previously out of reach of the QC method. Lastly, we present the key components of a high-performance, distributed-memory realization of the new method, including a novel algorithm for supporting unparalleled levels of deformation. Overall, this new formulation and implementation allows us to efficiently perform simulations containing an unprecedented number of degrees of freedom with low approximation error.

Contents

Acknowledgments	iii
Abstract	iv
1 Introduction	1
1.1 The behavior of metals at different scales	3
1.2 State of the art in material modeling	5
1.3 Quasicontinuum	6
1.4 Nanoscale structures and free surfaces	7
1.5 Scope and structure of this thesis	9
1.5.1 Novel, optimal summation rules	9
1.5.2 High performance implementation	11
1.5.3 Structure	11
2 Fully-nonlocal quasicontinuum formulation	13
2.1 The quasicontinuum method	13
2.1.1 Quasicontinuum approximation of an atomic ensemble	13
2.1.2 Summation Rules and Sampling Atoms	17
2.1.2.1 Consistent repatom masses	18
2.1.3 Spurious force artifacts	19
2.1.4 Example: Embedded Atom Method	21
2.2 A general framework for summation rules	21
2.2.1 Sampling atom weights	24
2.2.1.1 Zeroth- and first-order consistency	26
2.3 Analysis of summation rules	28
2.3.1 Analysis of summation rules in one dimension	28
2.3.1.1 Energy approximation error	28
2.3.1.2 Residual forces	34
2.3.1.3 Optimal summation rule	38
2.3.2 Analysis of summation rules in two dimensions	40

2.3.2.1	Force artifacts in two dimensions	40
2.3.2.2	Optimal summation rule in two dimensions	45
2.3.2.3	Optimal summation rule of second order in two dimensions	48
2.3.2.4	Residual force at a sharp mesh interface in two dimensions	51
2.3.2.5	Cauchy-Born kinematics in the nonlocal framework	57
2.3.2.6	Derivation of the average deformation for a discrete patch of simplicial elements	58
2.3.3	Summary and conclusions	59
2.4	Extension to three dimensions and the full range of element sizes	60
2.5	Summary and conclusions	64
3	Verification and Validation	65
3.1	Overview	65
3.2	Examinations of isolated configurations	67
3.2.1	Energy approximation errors in 2D mesh interfaces	67
3.2.2	Residual forces in 2D mesh interfaces	74
3.2.3	Residual forces in 3D mesh interfaces	77
3.2.4	Summary of examinations of isolated configurations	79
3.3	Elastic benchmark simulations in three dimensions	80
3.3.1	Local position and energy errors in a nano-void simulation	80
3.3.1.1	Effect of residual force removal	87
3.3.2	Elastic constants in randomly coarse-grained copper single crystals	88
3.4	Inelasticity and the formation of microstructure	95
3.4.1	2D nanoindentation	97
3.4.2	Large-scale 3D nanoindentation	99
3.5	Free Surfaces	100
3.5.1	Calculation of surface energies	105
3.5.2	Elastic modulus of a nano-rod	108
3.5.3	Plate with a cylindrical hole	111
3.6	Summary and conclusions	112
4	Implementation	117
4.1	General overview	117
4.1.1	Scope	117
4.1.2	Outline and data structures	118
4.2	Computing forces and equilibrating	119
4.2.1	Solver	120

4.2.2	Repatom weights	121
4.3	Key data structures	122
4.3.1	Mesh	122
4.3.1.1	Vacancies and nonconvexity	124
4.4	Distributed-memory parallelization	128
4.5	Weights for optimal summation rules	132
4.5.1	First-order weights in 2D	133
4.5.2	Second-order weights in 2D	136
4.5.3	Optimal summation rule weights in 3D	138
4.6	High deformation adaptive neighborhoods	139
4.7	Polycrystals	142
4.8	Remeshing	143
4.8.1	Global remeshing	144
4.8.2	Local remeshing	146
4.8.3	Discussion	151
4.8.4	Examples	151
4.8.4.1	Nanoindentation	153
4.8.4.2	Relaxation of a nano-cube	154
5	Discussion, Outlook, Conclusions	156
5.1	Achievements	156
5.2	Future challenges	158
Appendix A	High-deformation adaptive neighborhoods	161
A.1	Overview	161
A.2	What information is stored	163
A.3	Ensuring necessary neighborhoods under large deformation	164
A.3.1	Goal and purpose	164
A.3.2	High-level view of neighborhood generation	164
A.3.3	When to rebuild	165
A.3.3.1	Example failure rate	167
A.3.4	Undeformed and deformed cutoff radii	169
A.4	Neighborhood construction for different sampling rules	172
A.4.1	Cauchy-Born-type sampling atoms	172
A.4.2	Non-Cauchy-Born-type sampling atoms in regions of uniform crystal orientation	174
A.4.3	Non-Cauchy-Born-type sampling atoms in polycrystal simulations	175

Appendix B Repairing meshes	177
B.1 Overview	177
B.2 Three dimensional meshing and degenerate elements	177
B.3 Repairing algorithm and efficacy	181
B.4 Rounding point coordinates	184
B.5 Chosen meshing strategy	185
B.6 Ghost forces from mesh defects	187
Appendix C Parallelization	192
C.1 Background	192
C.2 Formulating QC for distributed memory parallelism	195
C.2.1 State decomposition	195
C.3 Parallel initialization of a QC simulation	197
C.4 Adaptive neighborhoods in parallel simulations	201

List of Figures

1.1 A fractal nanotruss structure.	1
1.2 The Eiffel tower as an example of a hierarchical truss structure.	2
1.3 Grains, grain boundaries, and defects.	3
1.4 An illustration of grain boundaries and twinning within metals.	4
1.5 The movement of an edge dislocation.	4
1.6 Dislocations propagate through materials, interact, and form dislocation networks.	5
2.1 Repatoms chosen from lattice sites.	15
2.2 Affine interpolation in the undeformed and deformed configurations.	16
2.3 Various types of summation rules introduced by the QC community: nodal, cluster, quadrature-type, and element quadrature rules.	22
2.4 Cluster summation.	23
2.5 Quadrature-type summation.	24
2.6 Tessellated weights for nodal and quadrature summation rules.	25
2.7 Overview of summation rules in one dimension.	29
2.8 Example of an interface in a one-dimensional chain of atoms.	30
2.9 Tessellated weights for quadrature summation rules.	32

2.10 Schematic view of a one-dimensional mesh interface separating elements of two different sizes.	34
2.11 Approximated energy of a periodic one-dimensional QC chain described by a Lennard-Jones potential vs. the displacement u of the central repatom for five different discretizations.	37
2.12 Schematic view of the new summation rule.	39
2.13 Example of a non-uniform patch of triangular elements that result in a residual force on the central repatom 0.	41
2.14 Quadrature-type Cauchy-Born neighborhoods in the optimal 2D summation rule. . .	47
2.15 Energy per lattice site for regular (top) and irregular (bottom) 2D QC meshes. . . .	49
2.16 Second-order new summation rule.	50
2.17 Two-dimensional scenario to determine the residual force on a sharp mesh interface. .	51
2.18 Schematic illustration of the optimal summation rule weight computation for arbitrary element sizes.	61
2.19 Illustration of sampling atom weights for the first-order and second-order rules. . .	62
3.1 Schematic illustration of the QC discretization and different types of summation rules. .	66
3.2 Poor sampling atom representation because of tessellation.	68
3.3 The setup of the 2D mesh interface energy approximation example.	69
3.4 Energy error per lattice site for sharp interfaces.	70
3.5 Energy error per lattice site for diffuse interfaces.	72
3.6 Average relative energy error per atom in the 2D QC meshes with sharp and diffuse interfaces.	73
3.7 Interface energy approximation error versus cost for various summation rules.	74
3.8 Meshes for the sharp and diffuse mesh interface examples.	75
3.9 Residual forces at sharp mesh interfaces in two dimensions for various summation rules.	76
3.10 Residual forces at diffuse mesh interfaces in two dimensions for various summation rules.	77
3.11 Residual forces in non-uniform QC meshes with sharp interfaces.	78
3.12 Residual forces in non-uniform QC meshes with diffuse interfaces.	79
3.13 Examples of single-crystalline cubes of pure copper with a nano-void.	81
3.14 Single-crystalline Cu nanocube with a spherical void, with sharp interfaces.	83
3.15 Single-crystalline Cu nanocube with a spherical void, with diffuse interfaces.	84
3.16 Comparison of errors of all lattice sites for nanocube with a void for five different summation rules and various interatomic potentials.	86

3.17	Effect of removing residual forces by dead loads.	88
3.18	Example repatom distributions in randomly coarse-grained crystals for four different repatom densities.	89
3.19	Example distributions of sampling atoms for four different summation rules applied to the same set of repatoms.	90
3.20	An illustration of repatom density and coarsening ratios.	90
3.21	Results of elastic constant calculations: errors and normalized computation times. .	91
3.22	Distribution of element sizes in the void-expansion simulation.	93
3.23	2D quasistatic nanoindentation example performed with automatic mesh refinement.	96
3.24	Indenter force vs. indentation depth for a two-dimensional Cu single crystal obtained from lattice statics and the QC method using the new second-order summation rule and automatic mesh adaptation.	97
3.25	Initial and final QC meshes of a nanoindentation simulation in two dimensions using automatic mesh adaptation.	98
3.26	Dislocation emission from the surface during nanoindentation simulated by lattice statics and the fully-nonlinear energy-based QC method with the new second-order summation rule and automatic mesh adaptation.	99
3.27	Force-displacement curves of many nanoindentation realizations for spherical and pyramidal indenters.	100
3.28	Indentation of a spherical indenter into a nano-cube of 86 billion lattice sites, represented by 16.8 million quasicontinuum degrees of freedom.	101
3.29	Zoomed-in snapshots of the microstructure evolving during 3D nanoindentation. . .	102
3.30	Schematic views of metallic free surfaces and pertinent QC summation rules.	103
3.31	Schematic view of the model used to determine surface energies.	105
3.32	Relative error of the computed surface energy vs. QC element size for single-crystalline (100) Cu.	106
3.33	Contour plots of the relative error of the surface energy as a function of the effective interaction radius r and the average element size L/a_0	107
3.34	Stress-strain response of nano-rods of different sizes modeled by the QC method with five different summation rules and three different average element sizes L	108
3.35	Deformed Cu nano-rods under uniaxial extension.	109
3.36	Size effects of the elastic modulus of a nano-rod.	110
3.37	QC models of a thin plate with a cylindrical hole with zoomed graphics showing the deformation of the surface-nearest atomic sites in the vicinity of the hole.	111
3.38	Finite element results for uniaxial loading, demonstrating the surface displacements near the hole and the von Mises stress distribution in the vicinity of the hole. . . .	112

3.39	Formation of dislocation loops and surface defects around the cylindrical hole. . . .	113
3.40	Reaction force for the scenario of a nanofilm with a cylindrical hole.	114
4.1	Mesh dependence in surface calculations.	123
4.2	Zero-volume element repair.	124
4.3	Degenerate elements and residual forces on a regular mesh.	125
4.4	Degenerate elements and residual forces on an irregular mesh.	126
4.5	Producing voids and vacancies.	127
4.6	Producing nonconvex geometry.	128
4.7	Nanoindentation domain decomposition summary.	130
4.8	Illustration of regions used to compute optimal summation rule weights.	134
4.9	2D example showing trimming planes and cuts to create a weight region for first-order optimal summation.	135
4.10	2D illustration of meshing optimal sampling atom weight regions to obtain weights.	136
4.11	2D example showing trimming planes and cuts to create a weight region for second-order optimal summation.	137
4.12	Dodecahedron approximation for 3D optimal summation weights.	138
4.13	3D example showing trimming planes and cuts to create a weight region for first-order optimal summation.	139
4.14	Example second-order weights for large and small elements, relative to the potential's interaction distance.	140
4.15	Example of the neighborhood of a sampling atom in the undeformed and deformed configurations during a nanoindentation simulation.	141
4.16	An example polycrystal domain.	142
4.17	Illustration of the longest edge propagation path refinement algorithm.	145
4.18	2D Lepp refinement example.	147
4.19	3D Lepp refinement example of a single element.	148
4.20	Existing local remeshing algorithm.	149
4.21	Refining to atomistic resolution with various 2D refinement algorithms.	152
4.22	Remeshing performed during a quasistatic 2D nanoindentation example with a pyramidal indenter.	153
4.23	The elements selected for refinement during a 3D nanoindentation example.	154
4.24	Relaxation of a free-standing nano-cube with adaptive mesh refinement	155
A.1	A severely-deforming 2D nanoindentation example illustrating the adaptive neighborhoods algorithm.	162

A.2	An example of the high deformation neighborhood algorithm rebuilding a neighborhood.	166
A.3	An example in which the neighborhood algorithm fails on free surfaces.	168
A.4	Neighborhood buffer distance illustration.	169
A.5	Illustration of deformed and undeformed cutoff radii using the neighborhood of a sampling atom experiencing severe deformation.	170
A.6	Another illustration of deformed and undeformed cutoff radii using the neighborhood of a sampling atom experiencing less deformation.	171
A.7	Illustrations of Cauchy-Born-type sampling atoms in elements of different sizes. . . .	173
A.8	An example of neighborhoods for polycrystal simulations.	175
B.1	Zero-volume element repair of two adjacent degenerate tetrahedra.	178
B.2	Repairing the sharp mesh from the Interface scenario 3.2.3	179
B.3	Repairing the diffuse mesh from the Interface scenario 3.2.3	180
B.4	Geometry that cannot be repaired.	181
B.5	Repairing the structured mesh from a scenario with two voids.	182
B.6	Repairing the random mesh from the Elastic Constants scenario 3.3.2	185
B.7	Reduction in zero-volume tetrahedra by rounding point coordinates.	186
B.8	Performance of repairing and rounding for various scenarios.	187
B.9	Ghost force arising from mesh defect.	188
B.10	Repairing removes the ghost force arising from the mesh defect.	189
B.11	Repairing from a different topology removes the ghost force arising from the mesh defect.	191
C.1	The process of parallel initialization.	198
C.2	Nanoindentation domain decomposition example.	200
C.3	Adaptive neighborhoods in parallel simulations.	202

List of Movies

1.1	Atoms initially distributed will rearrange to their lowest energy state. (link)	3
1.2	The formation and movement of dislocations under load. (link)	3
1.3	Grain boundaries block the movement of dislocations. (link)	4

3.4	Quasistatic 2D nanoindentation simulation with pyramidal indenter illustrating remeshing, sample atoms, and microstructure (link)	95
3.5	Dynamic 2D nanoindentation simulation with pyramidal indenter illustrating remeshing, sample atoms, and microstructure (link)	95
3.6	Quasistatic 2D nanoindentation simulation with spherical indenter illustrating remeshing, sample atoms, and microstructure (link)	98
3.7	Formation of dislocation loops and surface defects around a cylindrical hole in a nanofilm pulled in tension. (link)	112
4.8	FIRE solver iterations for a 2D quasistatic uniaxial extension simulation (link)	121
4.9	FIRE solver iterations for a 2D dynamic uniaxial extension simulation (link)	121
4.10	FIRE solver iterations for a 2D dynamic shear simulation (link)	121
4.11	2D example showing trimming planes and cuts to create a weight region for first-order optimal summation. (link)	135
4.12	2D example showing trimming planes and cuts to create a weight region for second-order optimal summation. (link)	136
4.13	3D example showing trimming planes and cuts to create a weight region for first-order optimal summation. (link)	138
4.14	A cut-through of the optimal summation weight regions in 3D. (link)	138
4.15	A cut-through of an example polycrystal domain. (link)	142
4.16	Illustration of the longest edge propagation path refinement algorithm. (link)	145
4.17	2D Lepp refinement example. (link)	147
4.18	3D Lepp refinement example of a single element. (link)	148
4.19	3D Lepp refinement example. (link)	149
4.20	Existing local remeshing algorithm choosing the split that minimizes the worst condition number. (link)	149
4.21	Refining to atomistic resolution with global remeshing (link)	152
4.22	Refining to atomistic resolution with local remeshing (link)	152
4.23	Refining to atomistic resolution with untruncated Lepp (link)	152
4.24	Refining to atomistic resolution with Lepp terminal level 2 (link)	152
4.25	The elements selected for refinement during a 3D nanoindentation example. (link) . .	153
4.26	An illustration of remeshing, adaptive neighborhoods, and implicit dynamics on a 2D nanoindentation problem (link)	153
A.27	A severely-deforming 2D nanoindentation example illustrating the adaptive neighborhoods algorithm. (link)	161
A.28	An example of the high deformation neighborhood algorithm rebuilding a neighborhood. (link)	165

A.29	An example in which the neighborhood algorithm fails on free surfaces. (link)	167
A.30	Illustration of deformed and undeformed cutoff radii using the neighborhood of a sampling atom experiencing severe deformation. (link)	169
A.31	Another illustration of deformed and undeformed cutoff radii using the neighborhood of a sampling atom experiencing less deformation. (link)	169
A.32	An example of neighborhoods for polycrystal simulations. (link)	175
C.33	Adaptive neighborhoods in parallel simulations. (link)	202

Chapter 1

Introduction

Many historians delineate human history by the materials from which humans make their tools. From stone to bronze and iron ages, improved materials facilitated advancement in all fronts of human civilization. Though the proper name of our current age is disputed (“steel”, “industrial”, “silicon”, and “plastic” are some possibilities), recent technological advances in nano-engineered materials indicate that perhaps the next age might be based on a new generation of functional materials which derive exceptional properties from features at micro- and nano-scales (see, for example [Biener et al., 2007, Schaedler et al., 2011, Montemayor et al., 2014]).

As manufacturing capabilities for systems with these nanoscale features become more readily available [Arnold, 2013], techniques to simulate their behavior are needed to inform the design process. Consider, for example, the fractal truss shown in Figure 1.1 (a nanotruss with a height less than 0.00005 meters [Jang et al., 2013]) and the Eiffel tower in Figure 1.2 (a truss with a height greater than 300 meters). Though both are hierarchical truss structures and bear superficial resemblance, their size differs by a factor of more than seven

million. Techniques for analyzing and predicting the behavior of structures such as the Eiffel tower date back to Claude-Louis Navier’s work in the early 1800s, are well-established, and modern engineering successfully leverages these techniques daily. However, the application of techniques for large-scale structures to micro- and nanoscale structures produces incorrect results.

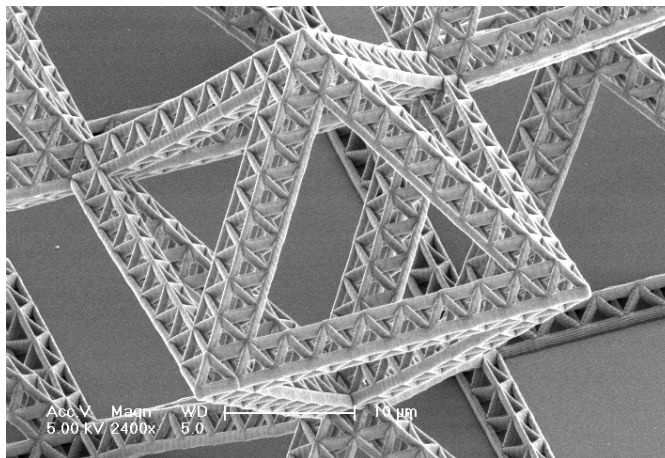


Figure 1.1: A fractal nanotruss structure. Truss members have walls with thickness on the order of nanometers. Image courtesy of Prof. Julia R. Greer at Caltech ([link](#))

(a) Eiffel tower ([link](#))(b) Truss detail ([link](#))(c) Truss members ([link](#))

(d) Individual trusses

Figure 1.2: The Eiffel tower as an example of a hierarchical truss structure. The overall frame is a truss structure (b) whose individual members are made of trusses themselves (c).

Techniques for large-scale structures depend on **the continuum hypothesis**, which is the idea that materials are “composed of an infinitely divisible continuous medium, imbued with a constitutive behaviour that remains unchanged regardless of how small the structure of interest may be.” [Miller and Tadmor, 2002] This assumption, though it conflicts with our knowledge that matter is composed of individual indivisible atoms, is valid for most practical contemporary engineering analysis and greatly reduces the work required to predict a system’s behavior.

The continuum hypothesis breaks down when a structure’s response to applied loads depends on effects at the atomic scale and those effects cannot be subsumed into a phenomenological model. These atomistic effects and nanoscale phenomena are precisely what govern the behavior of the promising engineered materials of the future and must be correctly modeled. Though it is possible to correctly capture these effects by simulating the behavior of a system of individual atoms, even the largest benchmark simulations of this type fall short of the required size by up to six orders

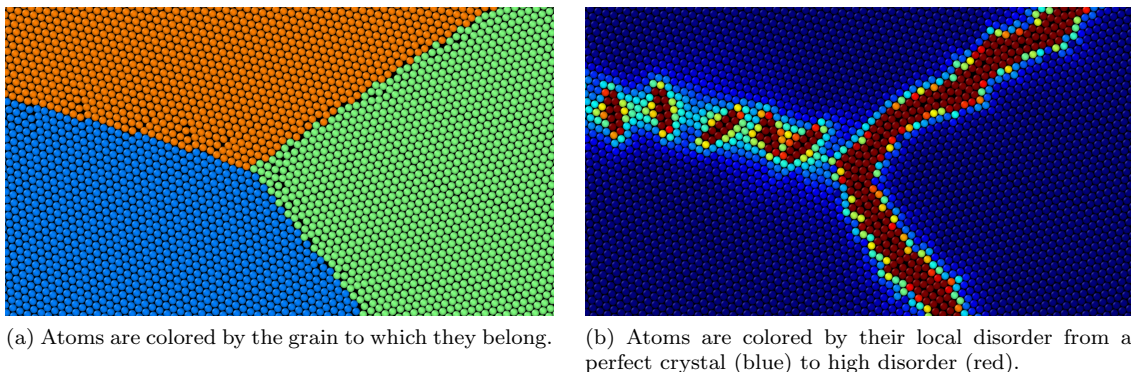


Figure 1.3: Grains, grain boundaries, and defects.

of magnitude because of limited computational resources. Though computer capabilities increase over time, physical limits on computer architecture imply that these simulations may never be possible. Understanding such phenomena requires modeling techniques that locally retain atomistic information while transitioning to the relevant macroscopic length scales where such information is not needed. The goal of this present work is to formulate a technique and create a high-quality efficient realization which can be used to investigate systems of engineering and scientific interest.

1.1 The behavior of metals at different scales

The strategy taken for a particular modeling technique depends on the underlying material; methods for predicting the behavior of amorphous metals can differ greatly from methods used for ceramics. The formulation and implementation within this thesis focus on modeling metals such as copper and aluminum.

Materials accommodate their applied loads and displacements by deformation, a process that is governed by the material's energy and entropy. Roughly speaking, a body in equilibrium seeks to minimize its total energy. Initially-disordered metallic atoms will attempt to rearrange themselves until they locally form a crystal of uniform orientation at equilibrium spacing, which is the lowest energy state for a metal (a movie illustrating this behavior is available [here](#)). Many grains (regions of uniform orientation) may form (see Figure 1.3) and the structure, density, and orientation of the boundaries between these grains influence the overall material response.

When metals are deformed, they experience a change in shape that can be recovered (elastic deformation) and, if they are deformed sufficiently, a change in shape which cannot be recovered (plastic deformation). Plastic deformation is the result of relative atomic movement and most commonly takes the form of slip and twinning (see Figure 1.4). The plasticity of slip is transmitted via dislocations, one type of which is illustrated in Figure 1.5. Under applied loads, these dislocations

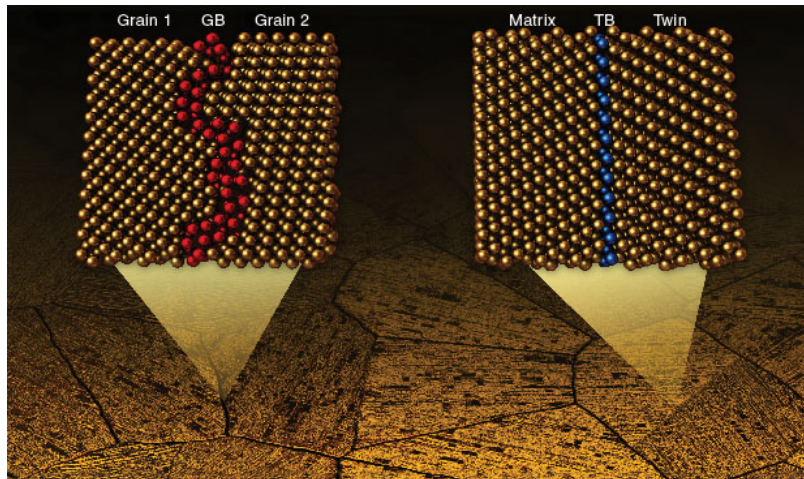


Figure 1.4: An illustration of grain boundaries and twinning within metals [Wang et al., 2013].

move through metals, interact, and form dislocation networks (see Figure 1.6) which can strengthen the material by inhibiting the movement of dislocations [Bulatov et al., 2006]. A movie illustrating the formation and movement of dislocations under load may be seen [here](#). Capturing the effect of these dislocations is critical when modeling structures at exceptionally small scales.

Most metallic systems of interest are polycrystalline, containing many grains. When polycrystal samples with large grains are deformed, grain boundaries block the propagation of dislocations and can strengthen the material (see movies [here](#) and [here](#) for illustrations of the interactions between dislocations and grain boundaries under load). However, below average grain sizes of about 10 – 15 nm, the deformation mechanisms change and the presence of grain boundaries weakens the overall sample response. This change in overall behavior stemming from microscale features is of great interest to the scientific community and forms a strong motivation for the development of modeling tools that can capture the behavior of dislocations within large polycrystal configurations.

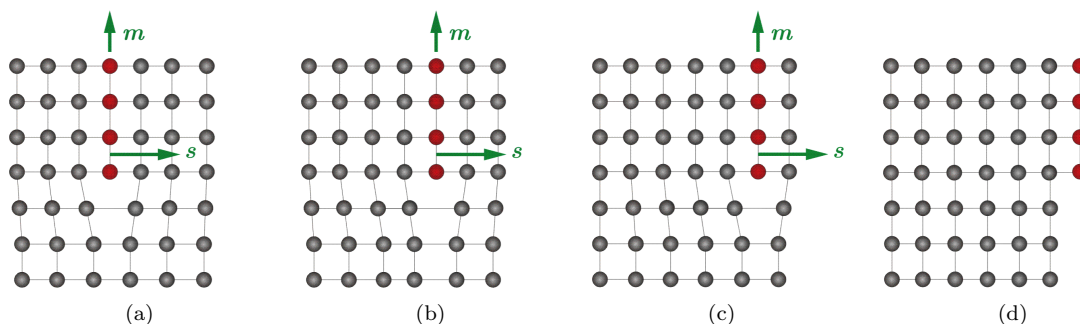


Figure 1.5: The movement of an edge dislocation, which transmits a relative atomic rearrangement through a material. Note that the dislocation leaves a perfect crystal in its wake (or at most a stacking fault in 3D).

Despite this rich behavior at the microscale originating from atomic-level rearrangements, most contemporary engineering analysis does not try to capture these effects but instead reduces the rules governing the behavior of metals to only a handful of numeric constants. The approximation is valid in most circumstances due to the sheer difference in length scales involved: dislocation cores govern response at the nanoscale and grains at the microscale, but most engineering applications use geometries six orders of magnitude larger than those features and the local atomic rearrangements can be averaged away.

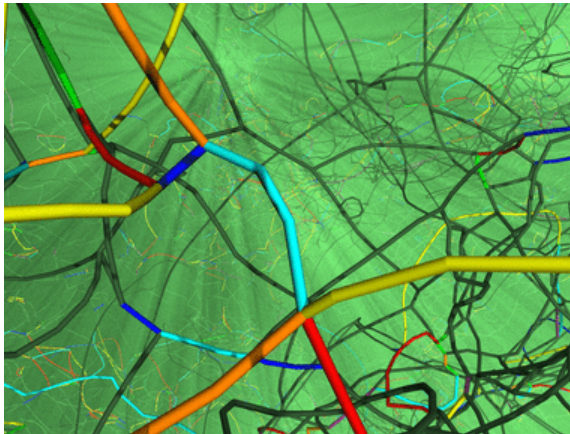


Figure 1.6: Dislocations propagate through materials, interact, and form dislocation networks.

1.2 State of the art in material modeling

The state of the art in material modeling offers highly-accurate methods for each individual length scale, from density functional theory (DFT) and molecular dynamics (MD) at the lower scales all the way up to continuum theories for the macroscale. Unfortunately, a wide gap exists due to a lack of models applicable at the mesoscale, where the continuum hypothesis fails yet atomistic techniques exceed computational limitations. Mastering this gap is key to understanding such diverse open problems as the response of nanoporous and nanostructured metals, the effective properties of nano- and microscale structures, or the underlying mechanisms of material failure. Therefore, there is urgent need for techniques that bridge across scales in order to accurately describe, thoroughly understand, and reliably predict the mechanics and physics of solids.

Various methods have been developed to bridge the scales. On the one hand, hierarchical schemes are the method of choice when a clear separation of scales can be assumed so that homogenization techniques can extract the effective constitutive response from the lower scales and pass it to the structural level. Examples include multiple-level finite-element analysis or FE^n [Miehe et al., 1999, Feyel and Chaboche, 2000, Schröder, 2000] as well as homogenization of atomistic ensembles at the material-point level of finite element simulations [Shephard et al., 2004, Chung, 2004, Park et al.,

2005, Clayton and Chung, 2006, Liu et al., 2006]. On the other hand, concurrent-scale coupling methods integrate different constitutive descriptions into a single-scale model by spatially separating domains treated, for example, by first-principles, MD, discrete defect mechanics, and continuum theories. Prominent examples comprise coupled atomistic/discrete-dislocation models such as CADD [Curtin and Miller, 2003, Shilkrot et al., 2004, Nair et al., 2010] and AtoDis [Brinckmann et al., 2012], as well as the bridging domain method [Belytschko and Xiao, 2003] and MADD [Abraham et al., 1998, Broughton et al., 1999], which couples several scales. In such methods, a key challenge is the passing of information across interfaces between different model domains.

Coarse-graining techniques circumvent this difficulty by applying the same lower-scale constitutive description to the entire model by up-scaling in space and/or in time. Examples include Coarse-Grained MD [Rudd and Broughton, 2005a,b] and the quasicontinuum (QC) method [Tadmor et al., 1996]. While spatial and temporal coarse-graining are equally important, we here focus on coarse-graining in space. The number of degrees of freedom is reduced by introducing geometric constraints, making the lower-scale accuracy efficiently available for larger-scale simulations; see [Tadmor et al., 1996, Izvekov and Voth, 2005, Suryanarayana, 2011, Iyer and Gavini, 2011] for examples. Coarse-graining in time (to transition from femtoseconds to minutes) can be added to the techniques investigated here but remains beyond the scope of this work; see for example [Voter, 1997, 1998, Sorensen and Voter, 2000, Voter et al., 2002, Kim et al., 2014a, Venturini et al., 2014]. Coarse-graining strategies offer a number of advantages over domain-coupling methods: (i) the model is solely based on the lower-scale constitutive description and hence comes with superior accuracy (in contrast to domain coupling methods, there is no need for an oftentimes phenomenological constitutive law in the coarse-grained domain); (ii) the transition from fully-resolved to coarse-grained is seamless (no approximate hand-shake region is required between different domains); (iii) depending on the chosen technique, model adaptation can efficiently reduce computational complexity.

1.3 Quasicontinuum

The *quasicontinuum* (QC) method was introduced to bridge from atomistics to the continuum by applying finite element interpolation schemes to atoms in a crystal lattice [Tadmor et al., 1996, Miller et al., 1998, Phillips et al., 1999]. This is achieved by three integral components: *geometric constraints* (which interpolate lattice site positions from the positions of a reduced set of representative atoms), *summation rules* (which avoid the computation of thermodynamic quantities from the full atomistic ensemble), and *model adaptation* schemes (which localize atomistic resolution and thereby efficiently minimize degrees of freedom). To date, numerous QC flavors have been developed which mainly differ in the choice of how the aforementioned three aspects are realized.

For example, geometric constraints in the original QC method [Tadmor et al., 1996, Shenoy et al.,

1998, Knap and Ortiz, 2001] were enforced by affine interpolation on a finite element mesh. More recent work has explored using higher-order shape functions [Park and Im, 2008, Kwon et al., 2009, Yang and To, 2015] as well as several formulations with meshless interpolation via smoothed-particle approaches [Yang and Xiao, 2005, Xiao and Yang, 2007, 2005, Wang and Guo, 2013, Xiao and Belytschko, 2005] or local maximum-entropy shape functions [Kochmann and Venturini, 2014]. Several extensions to allow for finite temperature (“hot-QC”) have been performed, including the combination of an effective energy function and Monte Carlo methods [Shenoy et al., 1999a], a generalization of QC that incorporates statistical mechanics to for constant-temperature simulations [Dupuy et al., 2005], the *max-ent* variational approach taken in [Kulkarni et al., 2008], an extension using Langevin dynamics [Marian et al., 2010], and more recent work that couples hot-QC with hyperdynamics to reduce the energy barriers between metastable states [Kim et al., 2014b] as well as [Venturini et al., 2014], in which the *max-ent* principle is coupled to mean-field models and kinetic laws. Recently there has been work applying the QC method to novel applications such as using multilattices for phase transforming materials [Dobson et al., 2007], and the simulation of beam lattices [Beex et al., 2014a,c]. Lastly, we note that many QC advances have been based on the published 2D QC code of Miller and Tadmor [Miller and Tadmor, 2011] and their website (qcmethod.org) serves as an invaluable hub for communication and coordination between QC researchers.

1.4 Nanoscale structures and free surfaces

Surface effects in nanoscale mechanical systems such as nanoporous solids or small-scale structures can have a significant impact on the effective material response which deviates from the material behavior of bulk solids. Understanding such phenomena requires modeling techniques that locally retain atomistic information while transitioning to the relevant macroscopic length scales, a requirement for which QC is well-suited. However, surfaces present a major challenge to coarse-grained atomistics, which has oftentimes been circumvented by costly ad-hoc extensions of the traditional QC method.

The mechanical behavior of nanoscale structures and nanostructured solids can deviate significantly from the material response observed in bulk solids. Example systems include micro- and nano-electro-mechanical devices and functional cellular solids with micro- and nano-sized morphological features, which promise tremendous technological and scientific potential. Nanoporous metals excel through unique physical, chemical, and mechanical characteristics [Biener et al., 2007], owing to the abundance of free surfaces. Depending on their degree of structural order, one distinguishes between nanoporous foams and structural materials (or, lattice materials). Both display high strength and stiffness and an extremely low mass density (as low as a few mg/cm³). In addition, lattice materials offer unprecedented opportunities for bottom-up-engineered materials [Schaedler et al.,

2011, Montemayor et al., 2014].

The extremely low spatial resolution of such systems is responsible for a variety of mechanical size effects. In particular, the abundance of free surfaces at the nanoscale alters the effective material behavior both within the reversible elastic regime and beyond. Surface relaxation alters the local atomic arrangement close to the surface, which in turn affects the atomic interactions and thus alters the effective elastic moduli. This size effect has been observed experimentally in single-crystalline nano-structures such as during elastic bending of nano-sized beams [Wong et al., 1997] or carbon nanotubes [Poncharal et al., 1999], as well as during tension-compression testing of nanowires; see for example [Chen et al., 2006, Agrawal et al., 2008, Asthana et al., 2011, Zhu et al., 2012, Chen et al., 2013]. Since surface relaxation is confined to a few atomic layers, the elastic size effects become noticeable only at small wire diameters on the order of a few nanometers, whereas the bulk response dominates at larger sizes.

Beyond the elastic limit surfaces interact with lattice defects, such as by acting as dislocation and vacancy sinks and sources. Remarkable increases of the tensile strength of single-crystal metallic whiskers with decreasing whisker diameter were first reported in the pioneering work of Taylor [1934] and later confirmed by Brenner’s studies (1956, 1957). Their research demonstrated what is now well-known as a structure-induced size effect in metals. Since then, there has been compelling experimental evidence that the strength of nano-sized single crystals exhibits a power-law dependence on the feature size [Zhu et al., 2009]. Recent tension-compression experiments on metallic micro- and nano-pillars impressively demonstrated this extrinsic size effect, cf. [Greer et al., 2009, Burek and Greer, 2010, Kim et al., 2011]. In particular, a variety of micro-deformation studies have revealed strong size effects in Cu single crystals [Dehm, 2009, Kiener et al., 2006, 2008, 2009, Maass et al., 2008, Jennings et al., 2010]. At such small scales, three mechanisms play an important role: small volumes reduce the number of defects in a statistical fashion, and surfaces interact with defects. The increase in strength has been explained by dislocation starvation [Greer et al., 2005, Shan et al., 2008, Greer and Nix, 2006], source exhaustion [Rao et al., 2008, Norfleet et al., 2008], source truncation [Parthasarathy et al., 2007, Rao et al., 2007], or weakest-link mechanisms [Norfleet et al., 2008]; see [Greer and De Hosson, 2011] for a review. In addition, surfaces alter the local stress state with important consequences for nanoscale structures; see for example [Gill, 2007, Pugno and Aifantis, 2011, Miri et al., 2011, Grekov and Yazovskaya, 2013].

From the modeling perspective, the study of nano-sized structural members is challenging. On the one hand, they reside outside the realm of traditional MD because computational costs severely limit the domain size that can be modeled as well as the range of viable strain rates [Derlet et al., 2003]. On the other hand, size effects in nanoscale structures as well as surface-defect interactions hardly admit the use of the continuum hypothesis which forms the basis of most engineering models. Nonlocal and size-aware continuum models have been proposed to capture individual size effects at

the nanoscale (see for example [Miller and Shenoy, 2000, Chen et al., 2006]), yet they are commonly tailored for particular mechanisms and not universally applicable. Furthermore, the specific length scales involved in nanoscale structures oftentimes prohibit a separation of atomistic and structural scales. This calls for a powerful multiscale simulation methodology that bridges across scales and is capable of modeling nano- to micrometer-sized objects at the accuracy of the underlying atomistic ensemble, a call to which the new quasicontinuum methodology developed in this thesis may be applied.

1.5 Scope and structure of this thesis

The primary contribution this thesis makes to the QC field is in a novel, fully nonlocal, energy-based QC formulation and the development and validation of new summation rules. The scheme and summation rules are analyzed in one and two dimensions, as well as numerically examined in three dimensions. The new summation rules are uniformly superior to existing summation rules in cost and efficacy, and in particular allow for an improved representation of free surfaces.

This thesis also makes significant contributions to the implementation of the QC method, which is able to examine unparalleled levels of deformation and unprecedented size (tens of millions of degrees of freedom). The computational cost of the method is examined, and data structures and algorithms are described. A novel algorithm for supporting the highly-deforming atomistic neighborhoods found in simulations with significant plasticity is developed and tested. The formulation is parallelized for a distributed-memory system using domain decomposition and message-passing, under the assumption that no single computer can hold the entire simulation's state at any given time. Lastly, the novel QC formulation and realization are tested in various verification and validation scenarios, including nanoindentation, void growth, and the effect of holes in nanofilms.

1.5.1 Novel, optimal summation rules

Though they were not originally seen as such, summation rules have become an essential component of recent analysis and development within the QC method. While the introduction of geometric constraints reduces the number of degrees of freedom, summation rules are required to efficiently reduce the number of lattice sites to be considered in the computation of, e.g., energies and forces. To this end, the thermodynamics of the full atomistic ensemble are approximated by those of a small set of carefully-chosen lattice sites (in the following referred to as *sampling atoms*), comparable to quadrature rules commonly used in the finite element method. To seamlessly bridge from atomistics to the continuum without differentiating between atomistic and coarse-grained regions, in this work we make use of a *fully-nonlocal QC* approximation without entire elements being represented by the Cauchy-Born rule, as is commonly done in local QC [Tadmor et al., 1996, Marshall and Dayal,

2013]. Summation rules now differ by the choice of (i) sampling atom locations and (ii) sampling atom weights.

Successful examples of summation rules introduced previously include node-based cluster summation [Knap and Ortiz, 2001] with sampling atoms located in clusters around repatoms, and quadrature-type summation [Gunzburger and Zhang, 2010, Yang et al., 2013] with sampling atoms chosen nearest to Gaussian quadrature points with or without the repatoms included as sampling atoms. Furthermore, element-based summation rules have been introduced in the nonlocal QC context [Gunzburger and Zhang, 2010, Yang et al., 2013] and have demonstrated superior accuracy over traditional cluster-based summation schemes [Iyer and Gavini, 2011]. Quite recently, an ad-hoc compromise between local and quadrature summation rules was proposed by Beex et al. [2014b], which is similar in spirit and contain as a special case within the new summation rules reported here. Many summation rules have been introduced in an ad-hoc manner to mediate QC deficiencies (for example, cluster rules were introduced to remove zero-energy modes in force-based QC [Knap and Ortiz, 2001]) or by borrowing schemes from related models (such as finite element quadrature rules [Gunzburger and Zhang, 2010]). The novel “optimal” summation rules presented in this work are derived with the goal of optimally integrating the energy of a system under the constraint of affine deformation within elements.

Summation rules can either approximate the energy of the system [Eidel and Stukowski, 2009] or the forces experienced by repatoms [Knap and Ortiz, 2001]. Systematic mathematical error and stability analyses of node-based cluster summation rules have highlighted the deficiencies of force-based QC schemes and have advocated the energy-based QC formulation [Luskin and Ortner, 2009, Dobson et al., 2010a,b,c] due to its conservative nature. Moreover, the proper convergence of energy-based QC techniques was shown recently for harmonic lattices [Espanol et al., 2013]. Therefore, in the following we choose an energy-based QC formulation. It is important to note that, although we focus on the QC method applied to atomistic ensembles, the same technique has been used in, e.g., coarse-grained density functional theory [Suryanarayana et al., 2013] and in multiscale structural mechanics [Beex et al., 2011]. As such, the presented results promise broad impact beyond the specific systems discussed here.

In this work, we combine all summation rules for energy-based QC into a unified scheme based on *sampling atoms* whose locations and weights differ between the various rules. In order to assess the accuracy of each scheme, we carry out a comprehensive analysis of energy approximation errors and resulting residual or spurious force artifacts in one and two-dimensions (going beyond all previous analytical studies that were primarily limited to 1D). Based on our findings, we introduce a class of summation rules with minimal approximation errors and marginal or vanishing residual forces in large elements and in the atomistic limit. We then use and compare the performance of various summation rules for representative simulations.

1.5.2 High performance implementation

An efficient, high-quality, robust and scalable realization of this novel QC formulation is necessary to simulate scenarios of engineering and scientific interest. Though several two-dimensional QC implementations exist, none provided the extensibility required for sampling atoms nor the performance required for the large, efficient simulations targeted by this work. The implementation created for this work supports 2D and 3D geometry and features a generic summation framework through which legacy and novel sampling schemes may be used. Many high-performance computing techniques are incorporated into the implementation to maximize its performance on a single computer, including cache-aware data structures and memory access patterns, automatic and manual vectorization, as well as threading and tuned load-balancing. However, a single computer does not have the capacity to store or simulate large domains, and as such the realization was formulated for use on distributed-memory machines where no single computer can hold the entire state of the simulation. This parallel implementation exhibits good weak-scaling to simulate very large system sizes (tens of millions of degrees of freedom) on even modest clusters and lays the groundwork for future simulations of unprecedented and unparalleled size.

To study the formation of microstructure, the implementation includes a novel adaptive technique to support neighborhoods that experience high deformation where sampling atoms must account for initially-distant neighbors. This neighborhoods scheme uses the relative movement of proximal neighbors to detect when local atomic rearrangements might cause new neighbors to enter the potential's cutoff radius and rebuilds the neighborhoods to account for the change. The algorithm is effective and efficient, maintaining correct neighborhoods during highly-deforming simulations without significantly increasing overhead.

1.5.3 Structure

This thesis is structured as follow: Chapter 2 first briefly introduces the quasicontinuum method, after which a general framework for summation rules is described. Errors are analyzed in one and two dimensions, the source of residual forces is derived, and optimal summation rules are constructed to eliminate spurious forces. Lastly, the new summation rules are extended to three dimensions.

In Chapter 3, simulations are performed and examined to explore the new formulation and summation rules. First, numerical experiments are performed to examine errors in the energy and forces within particular configurations of representative samples, without equilibration. Then, several integral tests are performed and comparisons are made directly to results obtained by MD as well as documented material properties. Lastly, because one of the important features of the new summation rules is the ability to correctly represent free surfaces, the superior performance of the new summation rules is confirmed via several examples with free surfaces.

Chapter 4 describes important aspects of the high performance implementation of the method used throughout the thesis. The general outline of the implementation is presented with its key data structures and algorithms. Parallelization for a distributed-memory system is described along with the challenges and constraints it imposes upon the formulation. The weights for the optimal summation rules of this formulation are easily described but implementation is unclear, and the approach used to produce the results in this thesis is enumerated for two and three dimensions. Finally, the novel adaptive neighborhood scheme and remeshing capabilities are reported.

Chapter 5 concludes the thesis with discussion and outlook on future work. The appendices provide more detail on certain topics of interest which are mentioned in the body of the thesis, including the neighborhood algorithm, the process used to repair meshes so that the novel summation rules can be used, and details related to the parallel formulation of the algorithm.

Chapter 2

Fully-nonlocal quasicontinuum formulation

In this chapter, we develop and analyze a novel fully-nonlocal energy-based quasicontinuum formulation. We introduce a framework that supports summation rules by including *sampling atoms*, or specific sites at which energetic quantities will be calculated. We investigate the origin of the spurious force artifacts termed *ghost forces* [Miller et al., 1998, Shenoy et al., 1998] and develop an optimal summation rule that greatly reduces these spurious forces for the same computation cost as existing summation rules. Lastly, we analyze these summation rules in one and two dimensions. Much of this chapter comes from a recent publication in the Journal of the Mechanics of Physics and Solids [Amelang et al., 2015], with some redactions and updated information.

2.1 The quasicontinuum method

2.1.1 Quasicontinuum approximation of an atomic ensemble

In classical mechanics, an atomistic ensemble containing N atoms is uniquely described by how their positions $\mathbf{q} = \{\mathbf{q}_1, \dots, \mathbf{q}_N\}$ and momenta $\mathbf{p} = \{\mathbf{p}_1, \dots, \mathbf{p}_N\}$ with $\mathbf{p}_i = m_i \dot{\mathbf{q}}_i$ evolve with time t . m_i is the mass of atom i , and dots denote material time derivatives. The system's Hamiltonian is defined by

$$\mathcal{H}(\mathbf{q}, \mathbf{p}) = \sum_{i=1}^N \frac{|\mathbf{p}_i|^2}{2m_i} + V(\mathbf{q}), \quad (2.1)$$

where the first term accounts for the kinetic energy and the second term represents the potential energy of the ensemble with V denoting a suitable atomic interaction potential. The time evolution of the system is given by Hamilton's equations, yielding Newton's equations of motion for atoms

$i = 1, \dots, N$, viz.

$$m_i \ddot{\mathbf{q}}_i = \mathbf{f}_i(\mathbf{q}) = -\frac{\partial V}{\partial \mathbf{q}_i}(\mathbf{q}), \quad (2.2)$$

where $\mathbf{f}_i(\mathbf{q})$ is the total force acting on atom i (the notion of conservative forces is implicit in the fact that forces are assumed to derive from a potential V). We note that V can, in principle, depend on both positions and momenta, which is necessary, e.g., when using the quasiharmonic approximation to mimic conditions of quasistatic finite temperature [Tadmor et al., 2013]. For concision, we here restrict our analysis to position-dependent potentials; the extension to include momenta is straightforward. Furthermore, we limit our analysis to crystalline solids in which the ground state atomic positions agree with sites of a Bravais lattice and the interaction potential most commonly has the additive decomposition

$$V(\mathbf{q}) = \sum_{i=1}^N E_i(\mathbf{q}) \quad (2.3)$$

with E_i being the energy of atom i . Note that some authors prefer to distinguish between external and internal forces by introducing external forces $\mathbf{f}_{i,\text{ext}}$ on atoms $i = 1, \dots, N$ so that

$$\mathbf{f}_i(\mathbf{q}) = \mathbf{f}_{i,\text{ext}}(\mathbf{q}_i) - \frac{\partial V}{\partial \mathbf{q}_i}(\mathbf{q}). \quad (2.4)$$

Ideally (and in any reasonable physical system), such external forces are conservative and derive from an external potential $V_{\text{ext}}(\mathbf{q})$ and we may combine internal and external forces into a single potential, which is tacitly assumed in the following. Examples of conservative external forces include gravitation, long-range Coulombic interaction, or multi-body interactions often realized by artificial potentials, see for example [Kelchner et al., 1998] for indentation.

Due to limitations of computational resources, it is generally not feasible to apply the above framework to systems that are sufficiently large to simulate long-range elastic effects, even for short-range interatomic potentials. However, except in the vicinity of lattice defects and flaws such as cracks or dislocations, the local environment of nearby atoms in a crystal lattice is almost identical up to translation. Therefore, the *quasicontinuum* (QC) *approximation* replaces the full atomistic ensemble by a reduced set of $N_h \ll N$ representative atoms (called *repatoms* throughout the remainder of this work, for brevity) having positions $\mathbf{x}(t) = \{\mathbf{x}_1(t), \dots, \mathbf{x}_{N_h}(t)\}$.

As an illustrative example, see Figure 2.1, where blue circles represent lattice sites that are chosen as repatoms. The displayed selection of repatoms might be useful if there were a lattice defect near the bottom left of the domain, while the rest of the domain simply experienced elastic deformation.

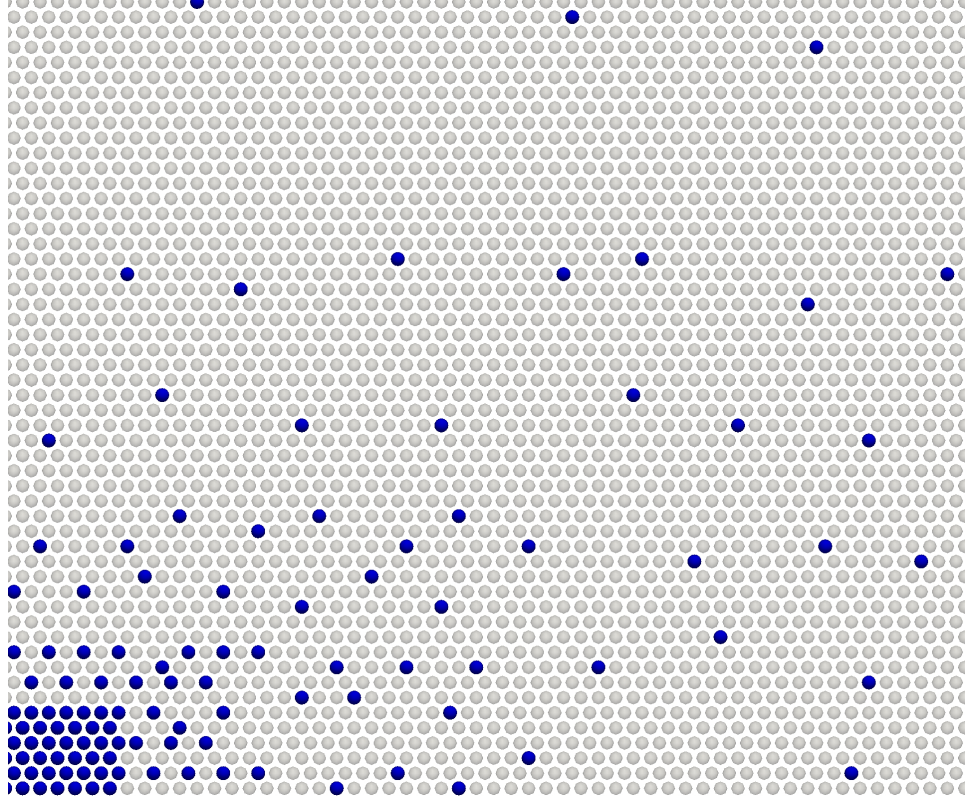


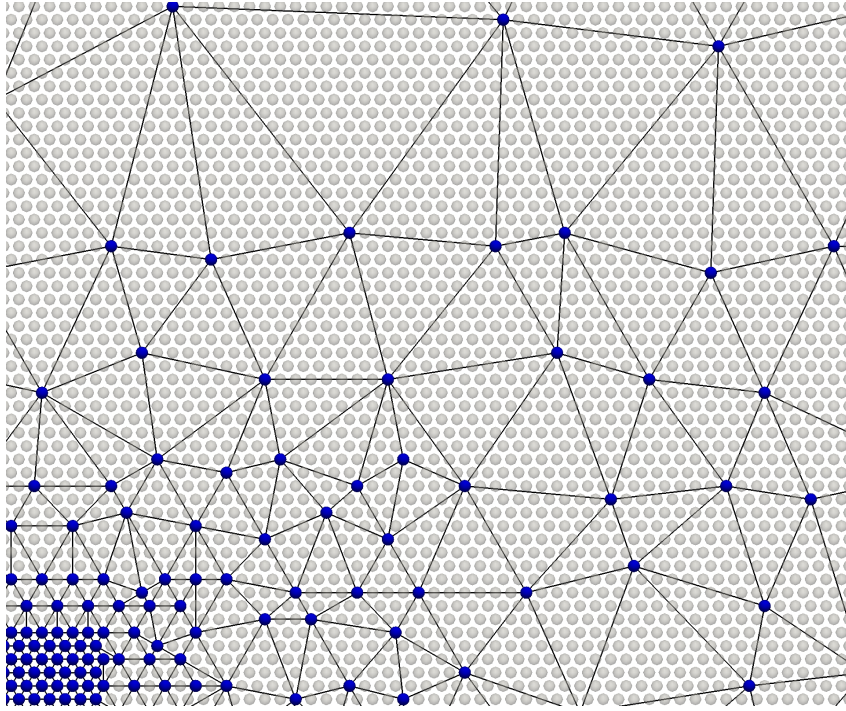
Figure 2.1: Repatoms chosen from lattice sites.

The approximate current position \mathbf{q}_i^h and momentum \mathbf{p}_i^h of atom i are obtained by interpolation:

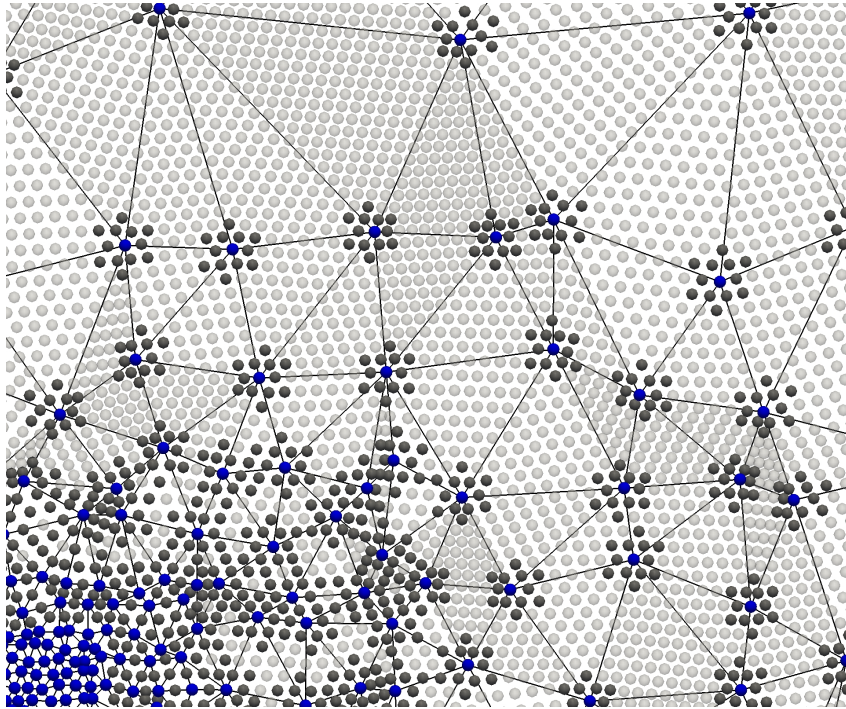
$$\mathbf{q}_i \approx \mathbf{q}_i^h = \sum_{a=1}^{N_h} N_a(\mathbf{X}_i) \mathbf{x}_a, \quad \mathbf{p}_i \approx \mathbf{p}_i^h = m_i \dot{\mathbf{x}}_i^h = m_i \sum_{a=1}^{N_h} N_a(\mathbf{X}_i) \dot{\mathbf{x}}_a \quad \text{for } i = 1, \dots, N. \quad (2.5)$$

$N_a(\mathbf{X}_i)$ is the shape function of repatom a evaluated at the position \mathbf{X}_i of lattice site i in the reference configuration. As an essential feature, this coarse-graining scheme locally recovers the exact atomic ensemble when all atoms are turned into repatoms (shape functions are chosen to satisfy the Kronecker property $N_a(\mathbf{X}_b) = \delta_{ab}$ for all $1 \leq a, b \leq N_h$ with δ_{ij} denoting Kronecker's delta). While, in principle, many different interpolation schemes may be used (see [Kochmann and Venturini, 2014] for meshless interpolation), in this work we will restrict our analysis to an affine interpolation on a Delaunay triangulated mesh as in the traditional QC method (as illustrated in Figure 2.2b).

The introduction of geometric constraints (2.5) has reduced the total number of independent degrees of freedom from $d \times N$ in d dimensions to $d \times N_h$. Consequently, the reduced Hamiltonian \mathcal{H}^h of the coarse-grained system, now involving approximate positions $\mathbf{q}^h = \{\mathbf{q}_1^h, \dots, \mathbf{q}_N^h\}$ and momenta



(a) Mesh in the undeformed configuration, connecting repatoms.



(b) Affine interpolation of lattice sites in the deformed configuration. Neighborhoods of repatoms are shown in grey.

Figure 2.2: Affine interpolation in the undeformed and deformed configurations.

$\mathbf{p}^h = \{\mathbf{p}_1^h, \dots, \mathbf{p}_N^h\}$, only depends on positions and momenta of the repatoms through (2.5):

$$\mathcal{H}^h(\mathbf{x}, \dot{\mathbf{x}}) = \sum_{i=1}^N \frac{|\mathbf{p}_i^h|^2}{2m_i} + V(\mathbf{q}^h). \quad (2.6)$$

Instead of solving for the positions and momenta of all N lattice sites, the QC approximation allows us to update only the positions and momenta of the N_h repatoms, which requires the computation of forces on repatoms. These are obtained from the potential energy by differentiation, which yields the force on repatom k :

$$\mathbf{F}_k(\mathbf{x}) = -\frac{\partial V(\mathbf{q}^h)}{\partial \mathbf{x}_k} = \sum_{j=1}^N \mathbf{f}_j^h(\mathbf{q}^h) N_k(\mathbf{X}_j) \quad \text{with} \quad \mathbf{f}_j^h(\mathbf{q}^h) = -\frac{\partial V(\mathbf{q}^h)}{\partial \mathbf{q}_j^h} = -\sum_{i=1}^N \frac{\partial E_i(\mathbf{q}^h)}{\partial \mathbf{q}_j^h}. \quad (2.7)$$

$\mathbf{f}_j^h(\mathbf{q}^h)$ denotes the total force acting on atom j . A common approximation for non-conservative external forces is

$$\mathbf{F}_{k,\text{ext}}(\mathbf{x}) = \sum_{j=1}^N \mathbf{f}_{j,\text{ext}}(\mathbf{q}^h) N_k(\mathbf{X}_j), \quad (2.8)$$

so that the overall force on repatom k can be written as

$$\mathbf{F}_k(\mathbf{x}) = \sum_{j=1}^N \left(\mathbf{f}_{j,\text{ext}}(\mathbf{q}^h) - \frac{\partial V(\mathbf{q}^h)}{\partial \mathbf{q}_j^h} \right) N_k(\mathbf{X}_j) = \sum_{j=1}^N \mathbf{f}_j^h(\mathbf{q}^h) N_k(\mathbf{X}_j), \quad (2.9)$$

$$\mathbf{F}_k(\mathbf{x}) = \sum_{j=1}^N \left(\mathbf{f}_{j,\text{ext}}(\mathbf{q}^h) - \frac{\partial V(\mathbf{q}^h)}{\partial \mathbf{q}_j^h} \right) N_k(\mathbf{X}_j) = \sum_{j=1}^N \mathbf{f}_j^h(\mathbf{q}^h) N_k(\mathbf{X}_j), \quad (2.10)$$

where $\mathbf{f}_j^h(\mathbf{q}^h)$ is the total force on atom j . Note that in an infinite Bravais lattice in the absence of external loading the forces on all atoms vanish (i.e., $\mathbf{f}_i^h(\mathbf{q}^h) = \mathbf{0}$), so that the forces on all repatoms vanish as well ($\mathbf{F}_k = \mathbf{0}$). This implies that the kinematic QC approximation itself does not introduce any force artifacts on repatoms.

2.1.2 Summation Rules and Sampling Atoms

Even though the number of degrees of freedom has been reduced from $d \times N$ to $d \times N^h$ by selecting repatoms, the calculation of repatom forces still requires computing the forces between all N atoms and their neighbors located within each atom's radius of interaction. In principle, summations in (2.9) can be reduced to the support of the shape functions to reduce cost. However, the force on any lattice site for which a repatom's shape function is nonzero will still need to be calculated, which becomes prohibitively expensive (especially in large elements). Without reducing the number of lattice sites at which forces must be calculated and if each sampling atom has on average N_b

neighbors, the $\mathcal{O}(N \times N_b)$ operations become computationally not feasible for realistically-sized systems. Therefore, *summation rules* have been introduced to replace the sum over index i in (2.3) or (2.9) by a weighted sum over N_s carefully-chosen *sampling atoms*, which are the only lattice sites at which energetic quantities are actually calculated. Consequently, the total potential energy is approximated by

$$V(\mathbf{q}^h) = \sum_{i=1}^N E_i(\mathbf{q}^h) \approx \tilde{V}(\mathbf{q}^h) = \sum_{\alpha=1}^{N_s} w_{\alpha} E_{\alpha}(\mathbf{q}^h), \quad (2.11)$$

where w_{α} is the weight of sampling atom α . Physically, w_{α} defines the number of atoms represented by sampling atom α . Because the summation rule is applied to the Hamiltonian, (2.11) is commonly referred to as an *energy-based* summation rule. The force experienced by repatom k is obtained by differentiation in analogy to (2.7), i.e.,

$$\tilde{\mathbf{F}}_k(\mathbf{x}) = -\frac{\partial \tilde{V}(\mathbf{q}^h)}{\partial \mathbf{x}_k} = -\sum_{\alpha=1}^{N_s} w_{\alpha} \sum_{j=1}^{N_b} \frac{\partial E_{\alpha}(\mathbf{q}^h)}{\partial \mathbf{q}_j^h} N_k(\mathbf{X}_j). \quad (2.12)$$

Now, repatom force calculation has been reduced to $\mathcal{O}(N_s \times N_b)$ operations. The selection of sampling atoms and the calculation of sampling atom weights aims for a compromise between maximum accuracy (sampling at every lattice site, expressed as $N_s = N$ and $w_{\alpha} = 1$ within this scheme) and maximum efficiency (sampling at very few carefully chosen lattice sites, expressed as $N_s \ll N$ and $w_{\alpha} \gg 1$).

On the one hand, choosing large numbers of sampling atoms results in high accuracy. Yet the large number of lattice sites involved in the calculation of energies and repatom forces leads to tremendous computational expenses which may defeat the purpose of coarse-grained atomistics. On the other hand, choosing very few sampling atoms considerably speeds up calculations and allows for the simulation of micron-sized problems (and above). However, this can result in large approximation errors, especially if the sampling atom locations and/or weights are chosen poorly.

2.1.2.1 Consistent repatom masses

Although all subsequent examples will be restricted to quasistatics, we would like to point out that the aforementioned fully-nonlocal energy-based QC formulation provides consistent repatom masses, which are required, e.g., for finite-temperature simulations [Tadmor et al., 2013, Venturini et al., 2014]. The total kinetic energy of the atomic ensemble (approximated by the QC interpolation scheme) is given by

$$\frac{1}{2} \sum_{i=1}^N m_i (\dot{\mathbf{q}}_i^h)^2 = \frac{1}{2} \sum_{i=1}^N m_i \left| \sum_{a=1}^{N_h} N_a(X_i) \dot{\mathbf{x}}_a \right|^2 = \frac{1}{2} \sum_{a=1}^{N_h} \sum_{c=1}^{N_h} \dot{\mathbf{x}}_a \cdot \left(\sum_{i=1}^N m_i N_a(X_i) N_c(X_i) \right) \dot{\mathbf{x}}_c, \quad (2.13)$$

where the term in parentheses may be interpreted as components M_{ac}^h of a consistent mass matrix \mathbf{M}^h . In avoidance of solving the global system during time stepping, one may use particle-mass lumping, cf. [Knap and Ortiz, 2001]. Applying the energy-based summation rule to the kinetic energy (analogously to the approximation of the potential energy) yields a consistent approximation of the kinetic energy with the consistent mass matrix components

$$M_{ac}^h = \sum_{i=1}^N m_i N_a(X_i) N_c(X_i) \approx \widetilde{M}_{ac}^h = \sum_{b=1}^{N_s} w_b m_b N_i(X_b) N_j(X_b). \quad (2.14)$$

The spatial coarse-graining outlined here, in principle, admits the study of dynamic phenomena. Yet the dynamic QC formulation gives rise to new challenges associated with spurious wave reflections and refractions in non-uniform meshes which we will not address here.

2.1.3 Spurious force artifacts

Summation rules on the energy level not only reduce computational complexity but also give rise to *force artifacts*; see for example [Eidel and Stukowski, 2009] for a recent discussion. By rewriting (2.12) without neighborhood truncations as

$$\widetilde{\mathbf{F}}_k(\mathbf{x}) = - \sum_{\alpha=1}^{N_s} w_{\alpha} \sum_{j=1}^N \frac{\partial E_{\alpha}(\mathbf{q}^h)}{\partial \mathbf{q}_j^h} N_k(\mathbf{X}_j) = \sum_{j=1}^N \left(- \sum_{\alpha=1}^{N_s} w_{\alpha} \frac{\partial E_{\alpha}(\mathbf{q}^h)}{\partial \mathbf{q}_j^h} \right) N_k(\mathbf{X}_j), \quad (2.15)$$

and comparing with (2.9), we notice that the term in parentheses in (2.15) does not equal force \mathbf{f}_j^h and hence does not vanish in general even if $\mathbf{f}_j^h = \mathbf{0}$ for all atoms (unless the set of sampling atoms is chosen to agree with the full atomic ensemble, i.e., $N_s = N$ and $w_{\alpha} = 1$). As a consequence, the QC representation with energy-based summation rules shows non-physical *residual forces* on repatoms in the undeformed ground state. Note that for uniform QC meshes (i.e., uniform repatom spacings, uniform sampling atom distribution, and a regular mesh), these residual forces disappear due to symmetry because the sum in parentheses in (2.15) cancels pairwise when carrying out the full summation. Residual forces therefore only appear in spatially non-uniform meshes; see for example [Iyer and Gavini, 2011] for an overview.

In the original QC method [Tadmor et al., 1996] where atomistic and coarse-grained regions (treated by a local Cauchy-Born formulation) are spatially separated, such forces only appear at the interface between atomistic and continuum regions and were called *ghost forces* [Miller et al., 1998, Shenoy et al., 1998]. In nonlocal, energy-based schemes – such as the method discussed here – the analogous forces are known sometimes as “spurious” and sometimes as “residual” forces [Eidel and Stukowski, 2009].

In contrast to previous studies, we would like to explicitly differentiate between residual and

spurious forces. *Residual forces* denote force artifacts in the undeformed configuration which can be removed by computing those forces and subtracting them as dead loads from all repatom forces in subsequent calculations, similarly to the method of ghost-force correction; see for example [Shenoy et al., 1998]. However, force errors due to summation rules depend on repatom positions and generally increase away from the undeformed configuration. Removing *residual* forces from the forces calculated in anything but the undeformed configuration *does not remove* these force errors. Therefore, *spurious forces* arise, i.e., force artifacts in the deformed configuration that cannot be corrected for in a simple manner. Even though residual and spurious forces are conservative in the energy-based scheme, they are non-physical and can typically be quantified in defect-free infinite systems.

To avoid residual forces, Knap and Ortiz [2001] introduced *force-based summation rules* which do not approximate the Hamiltonian but instead approximate the repatom forces explicitly, i.e., the summation rule is applied directly to (2.7), giving

$$\mathbf{F}_k(\mathbf{x}) = \sum_{i=1}^N \mathbf{f}_i^h(\mathbf{q}^h) N_k(\mathbf{X}_i) \approx \tilde{\mathbf{F}}_k(\mathbf{x}) = \sum_{\alpha=1}^{N_s} w_\alpha \mathbf{f}_\alpha^h(\mathbf{q}^h) N_k(\mathbf{X}_\alpha). \quad (2.16)$$

Consequently, the force-based formulation does not produce any residual forces because $\mathbf{f}_\alpha^h(\mathbf{q}^h) = \mathbf{0}$ in an undeformed infinite crystal (they become apparent, though, when considering finite-sized crystals with free surfaces). Yet this approximation gives rise to a new problem: the force-based method is non-conservative; ergo, there is no energy potential from repatom forces derive. This has a number of drawbacks; see for example [Eidel and Stukowski, 2009, Miller and Tadmor, 2009, Iyer and Gavini, 2011]. In quasistatic problems, the non-conservative framework may lead to slow numerical convergence, cause numerical instability, or converge to non-physical equilibrium states. This has been confirmed recently by mathematical analyses highlighting numerical instability and loss of accuracy due to force sampling; see for example [Luskin and Ortner, 2009, Dobson et al., 2010a,b,c, Ortner, 2011]. Furthermore, for dynamic or finite-temperature scenarios, a QC approximation using force-based summation rules cannot be used to simulate systems in the microcanonical ensemble (where the system's energy is to be conserved). It should be noted that, of course, this does not rule out the use of alternative ensembles. Finally, repatom masses, required for dynamic simulations, are not uniquely defined because there is no effective kinetic energy potential when using force-based summation rules. In contrast, energy-based summation rules lead to conservative forces and to strictly-symmetric stiffness matrices. Moreover, these matrices exhibit only the six admissible zero eigenvalues even for purely nodal summation rules as shown by [Eidel and Stukowski, 2009]. For all these reasons, in this thesis we focus on energy-based summation rules.

2.1.4 Example: Embedded Atom Method

The above coarse-graining framework is sufficiently general for most types of crystalline solids. Let us exemplify the general concept by a frequently-used family of interatomic potentials for metals to be used in our numerical examples. The Embedded Atom Method (EAM) by [Daw and Baskes \[1984\]](#) defines the interatomic potential energy of a collection of N atoms by

$$E_i(\mathbf{q}) = \frac{1}{2} \sum_{\substack{j=1 \\ j \neq i}}^N \Phi(r_{ij}) + \mathcal{F}(\rho_i), \quad \rho_i = \sum_{j=1}^N f(r_{ij}). \quad (2.17)$$

The pair potential $\Phi(r_{ij})$ represents the energy due to electrostatic interactions between atom i and its neighbor j , whose distance is given by $r_{ij} = |\mathbf{r}_{ij}|$ and $\mathbf{r}_{ij} = \mathbf{q}_i - \mathbf{q}_j$. ρ_i denotes the (approximated) effective electron density which atom i senses due to its neighboring atoms. $f(r_{ij})$ is the electron density at site i due to atom j as a function of their distance r_{ij} . $\mathcal{F}(\rho_i)$ accounts for the energy release upon embedding atom i into the local electron density ρ_i . From (2.2), the exact force \mathbf{f}_k acting on atom k is obtained by differentiation, viz.

$$\mathbf{f}_k(\mathbf{q}) = - \sum_{i=1}^N \frac{\partial E_i(\mathbf{q})}{\partial \mathbf{q}_k} = - \sum_{j \in n_I(k)} [\Phi'(r_{kj}) + \{\mathcal{F}'(\rho_k) + \mathcal{F}'(\rho_j)\} f'(r_{kj})] \frac{\mathbf{r}_{kj}}{r_{kj}}. \quad (2.18)$$

Since most non-ionic metallic potentials are short-range, we efficiently truncated the summations to include only neighboring atoms within the radius of interaction ($n_I(i)$ denotes the set of relevant neighbors of atom i). Introducing the QC approximation along with an energy-based summation rule now reduces the approximate repatom forces to [\[Eidel and Stukowski, 2009\]](#)

$$\tilde{\mathbf{F}}_k(\mathbf{x}) = - \sum_{\alpha=1}^{N_s} w_\alpha \sum_{j \in n_I(\alpha)} \left[\frac{1}{2} \Phi'(r_{\alpha j}^h) + \mathcal{F}'(\rho_\alpha^h) f'(r_{\alpha j}^h) \right] \frac{\mathbf{r}_{\alpha j}^h}{r_{\alpha j}^h} [N_k(\mathbf{X}_\alpha) - N_k(\mathbf{X}_j)], \quad (2.19)$$

which display residual and spurious forces as demonstrated in subsequent sections. Note that in the following we neglect thermal fluctuations and consider only zero temperature as in the original QC method. For finite-temperature QC extensions see for example [\[Shenoy et al., 1999a, Hai and Tadmor, 2003, Dupuy et al., 2005, Tang et al., 2006, Kulkarni et al., 2008, Marian et al., 2010, Tadmor et al., 2013, Venturini et al., 2014\]](#) and references therein. These approaches can be applied to the spatial coarse-graining techniques discussed here, yet they go beyond the scope of this thesis.

2.2 A general framework for summation rules

We consider energy-based summation rules applied to a crystalline lattice that is coarse-grained by the QC approximation using an affine interpolation. The total potential energy is approximated

by (2.11) and repatom forces by (2.12). Specific summation rules differ by the selection of the N_s sampling atoms and by the calculation of their weights w_α . The choice of sampling atoms is, in principle, independent of the choice of repatoms. Figure 2.3 schematically illustrates four types of summation rules previously introduced by the QC community.

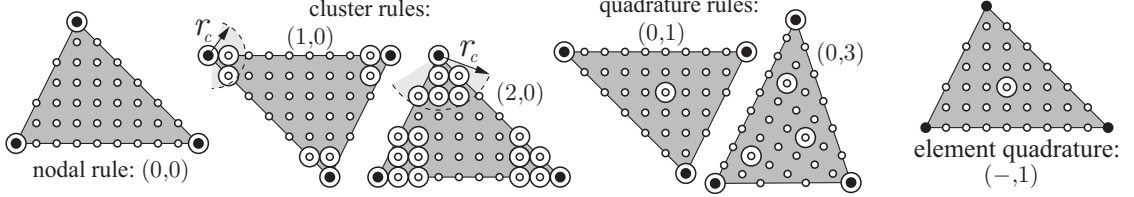


Figure 2.3: Various types of summation rules introduced by the QC community: nodal, cluster, quadrature-type, and element quadrature rules. Small open circles represent lattice sites, solid black circles are repatoms, and open large circles are sampling atoms. Except in element quadrature, all summation rules include the repatoms in the set of sampling atoms in order to naturally bridge to full atomistics.

In the *nodal* summation rule the set of sampling atoms agrees with the set of repatoms ($N_s = N_h$). In avoidance of the significant energy approximation errors and resulting force artifacts, *cluster* summation rules were introduced which choose sampling atoms as lattice sites located in spherical clusters around repatoms [Knap and Ortiz, 2001, Eidel and Stukowski, 2009]. Figure 2.4 shows cluster summation for the illustrative example; repatoms are blue and sampling atoms are red. The nodes, which are both repatoms and sampling atoms, are colored blue. Increasing the cluster radius r_c to include more shells of atoms increases the accuracy but also significantly raises computational expenses since the number of atoms per cluster scales with $O(r_c^3)$. Eidel and Stukowski [2009] showed that to reduce errors and force artifacts to a reasonable level clusters must be chosen sufficiently large to overlap, which incurs high computational expenses.

The main deficiency of all node-based summation rules is the error made by approximating the energy of many lattice sites well within an element by those at or near the vertices. As a remedy, *element quadrature* summation rules do not use sampling atoms on the nodes, but solely take sampling atoms as those lattice sites closest to quadrature points of the finite element mesh using, e.g., Gauss-Lobatto quadrature [Gunzburger and Zhang, 2010, Yang et al., 2013]. As a special case, the local QC formulation of Tadmor et al. [1996] can also be understood as an element quadrature rule where instead of using actual atomic neighbors the Cauchy-Born rule is imposed.

Quadrature-type summation rules sample over all repatoms plus quadrature-type lattice sites comparable to stress-point integration in particle methods [Xiao and Yang, 2007]. Figure 2.5 shows quadrature-type summation for the illustrative example. Quadrature sampling atoms are chosen on the lattice sites closest to the traditional Gauss-Lobatto points of a given order. In addition, one can formulate mixed summation rules having clusters around repatoms or quadrature-type sites, or around both [Gunzburger and Zhang, 2010]. Further summation rules have been proposed which

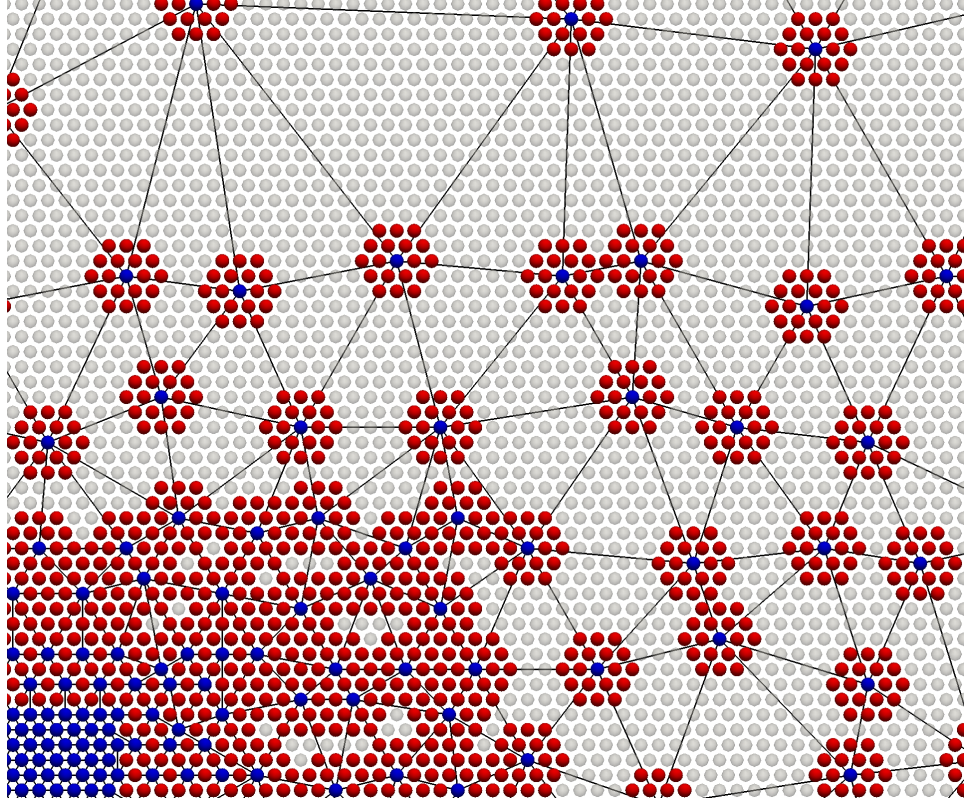


Figure 2.4: Cluster summation: repatoms are blue and sampling atoms are red. The nodes, despite being colored blue, are both repatoms and sampling atoms.

identify unique atomic neighborhoods for all lattice sites within a given element and compute the exact energy by weighting their energies [Yang et al., 2014]. While this approach effectively removes force artifacts, it is computationally much more expensive due to large numbers of sampling atoms and it is difficult to generalize since each geometrically-different element requires, in principle, a different summation rule. We therefore exclude this method from our analysis.

For brevity, we identify summation rules by two numbers: n_{Cl} , the number of shells included in clusters around repatoms, and n_Q , the number of quadrature-type sampling atoms per element. In principle, one could use clusters around quadrature points as well, but we do not consider summation rules of that type for reasons that will be explained later. Hence, a summation rule is characterized by the pair (n_{Cl}, n_Q) , cf. Figure 2.3. To seamlessly bridge to full atomistics, we will not conceptually differentiate between local and nonlocal elements. Therefore, we exclude element quadrature rules of type $(-, n_Q)$ from our numerical examples because they do not include repatoms in the set of sampling atoms so that some assumptions and a special algorithmic treatment may be required to recover full atomistics. All other schemes can transition seamlessly without explicitly differentiating between atomistic and coarse-grained regions by ultimately having all lattice sites be repatoms and sampling atoms.

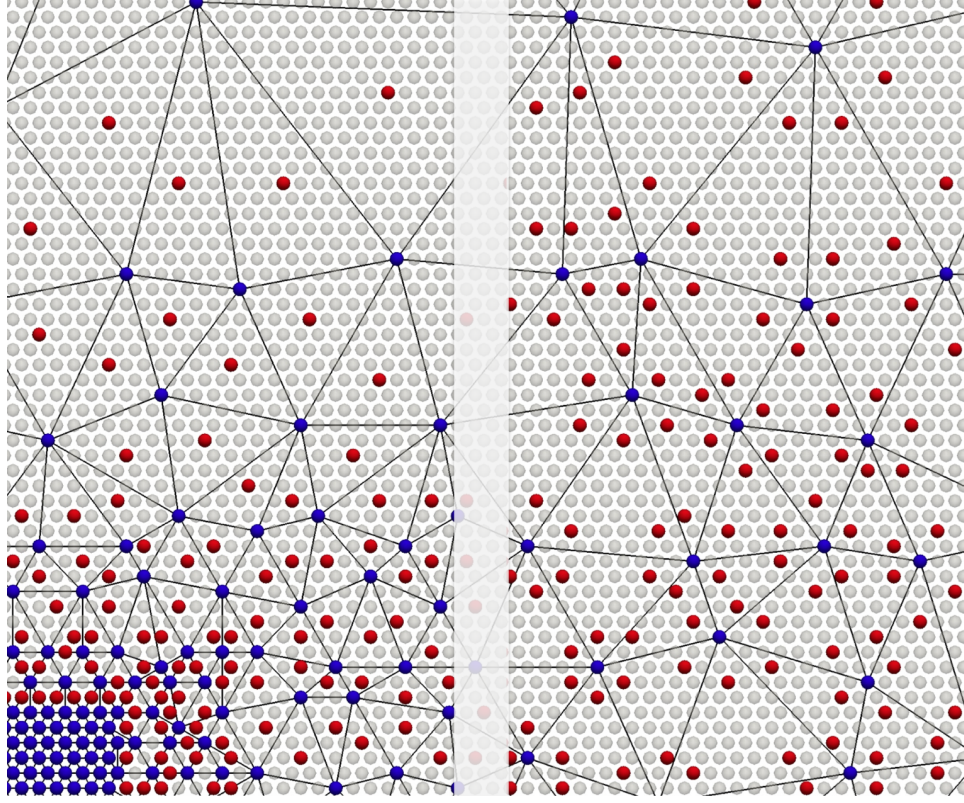
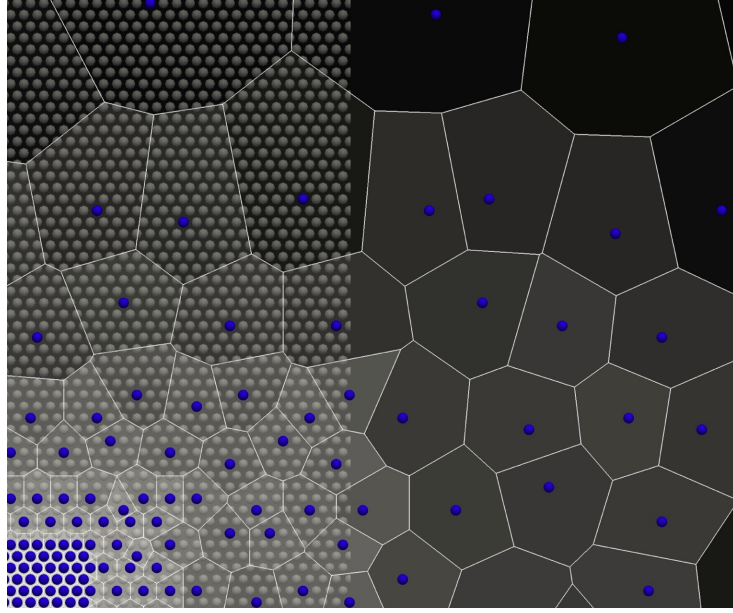


Figure 2.5: Quadrature-type summation: repatoms are blue and sampling atoms are red. Left panel shows one point quadrature $(0, 1)$, right panel shows three point quadrature $(0, 3)$.

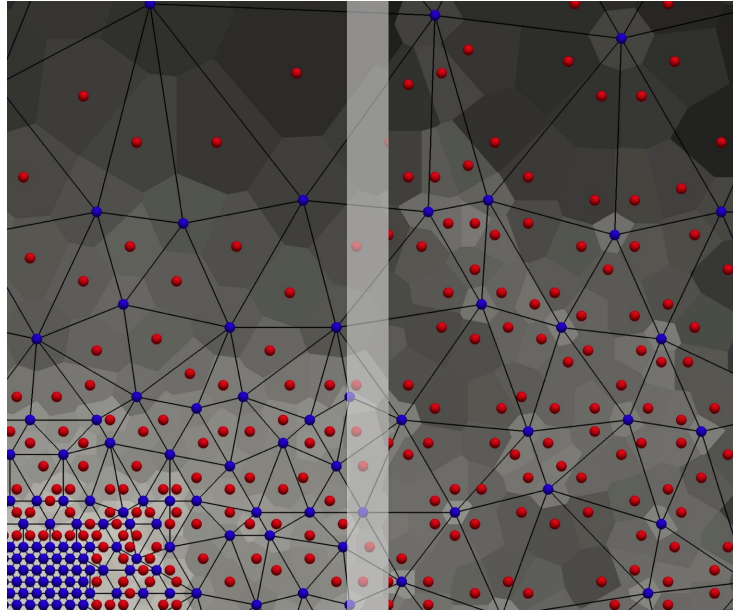
We specifically focus on quadrature-type summation rules which have been neglected in numerical QC realizations in the past. [Gunzburger and Zhang \[2010\]](#) reported that quadrature-type summation has, for the same accuracy, lower complexity than cluster-based approximation schemes. Similarly, [Luskin and Ortner \[2009\]](#) reported that cluster summation rules (both force and energy-based) lead to inconsistent and inaccurate QC approximations when used with non-uniform meshes and that increasing the cluster size does not resolve this problem. However, element-based summation rules (such as using clusters around quadrature points [Gunzburger and Zhang \[2010\]](#)), though not passing the patch test, show significantly smaller errors than node-based cluster summation rules, which decay with element size for sufficiently large elements [[Iyer and Gavini, 2011](#)].

2.2.1 Sampling atom weights

In most previous schemes, weights have been computed by solving a more or less complicated system of equations to satisfy two main requirements. First, in the full atomistic limit ($N_h \rightarrow N$) we must recover the correct energy, i.e., we must have $w_\alpha \rightarrow 1$. Second, weights were chosen such that the summation rule is zeroth- and first-order consistent to guarantee convergence with mesh refinement. Here, we propose to compute weights w_α from the volumes Ω_α of Voronoi cells in a tessellation based



(a) Nodal Summation Rule



(b) Quadrature Summation Rules

Figure 2.6: Tessellated weights for nodal and quadrature summation rules.

on the set of all sampling atoms, i.e.,

$$w_\alpha = \rho \Omega_\alpha \quad (2.20)$$

with ρ being the effective atomic density (which is uniformly constant for a pristine single crystal). By defining $\rho = 1/\Omega_0$ with Ω_0 the Voronoi cell volume of each atom in an undeformed perfect single

crystal and by choosing an interpolation scheme that satisfies the Kronecker property $N_a(\mathbf{x}_b) = \delta_{ab}$ (such as affine interpolation), we ensure the correct atomistic limit.

Weights computed from Voronoi tessellation are illustrated in Figure 2.6. On the left side of Figure 2.6a, the lattice sites are shown with the Voronoi cells superimposed. On the right side of Figure 2.6a, each sample is assigned a weight from the number of lattice sites inside of each Voronoi cell. Figure 2.6b shows tessellated weights for quadrature, where the Voronoi cells are shaded by their volume and the mesh is superimposed.

Computing weights in this way bears a number of advantages: (i) the computation of weights is computationally feasible for any spatial distribution of sampling atoms and does not require further approximations (such as differentiating between small and large elements); (ii) the computation of weights is independent of the chosen interpolation scheme and is therefore generally applicable, for example to meshless interpolations [Kochmann and Venturini, 2014]; (iii) the resulting summation rules automatically satisfy zeroth- and first-order consistency exactly (approximately, though, in the case of cluster rules).

2.2.1.1 Zeroth- and first-order consistency

The satisfaction of zeroth- and first-order consistency is shown as follows. To assess the consistency of a summation rule with Voronoi-tessellated weights, we introduce an energy E which varies linearly, giving rise to atomic energies $E_i = E(\mathbf{X}_i) = E_0 + \mathbf{E}_1 \cdot \mathbf{X}_i$. The exact total energy is

$$\sum_{i=1}^N E_i = N(E_0 + \mathbf{E}_1 \cdot \mathbf{X}_s), \quad \text{where} \quad \mathbf{X}_s = \frac{1}{N} \sum_{i=1}^N \mathbf{X}_i \quad (2.21)$$

denotes the center of mass of the lattice containing N sites. For a summation rule with weights (2.20) we have

$$\sum_{i=1}^N E_i \approx \sum_{\alpha=1}^{N_s} w_{\alpha} E_{\alpha} = E_0 \left(\sum_{\alpha=1}^{N_s} w_{\alpha} \right) + \mathbf{E}_1 \cdot \left(\sum_{\alpha=1}^{N_s} w_{\alpha} \mathbf{X}_{\alpha} \right). \quad (2.22)$$

For the summation rule to be exact up to first order, we hence must have

$$\sum_{\alpha=1}^{N_s} w_{\alpha} = N, \quad (2.23a)$$

$$\frac{1}{N} \sum_{\alpha=1}^{N_s} w_{\alpha} \mathbf{X}_{\alpha} = \mathbf{X}_s. \quad (2.23b)$$

Condition (2.23a) is satisfied automatically when choosing weights according to (2.20) because the Voronoi tessellated volumes span the entire body and the density is exact. This is true even for sampling atoms on the surface if we truncate Voronoi volumes conformingly [Kochmann and Venturini,

2014]. Condition (2.23b) is also satisfied by (2.20) for any distribution of sampling atoms. This can easily be verified by a geometric argument; consider two nearby sampling atoms: the linear field along the line between the two Voronoi centers is summed exactly because the two weights along the line are identical. Since this argument holds for any pair of nearby sampling atoms, the linear field is summed exactly. In conclusion, the chosen weights satisfy zeroth- and first-order consistency, which promotes convergence with h -refinement.

For cluster summation rules we assign a unique weight to each cluster, i.e., we choose the $N_{s,\text{Cl}}$ cluster center atoms, and each cluster center α is surrounded by a set $n_C(\alpha)$ of cluster atoms. Then, the total potential energy becomes

$$\tilde{V}(\mathbf{x}) = \sum_{\alpha=1}^{N_{s,\text{Cl}}} w_{\alpha} \sum_{\beta \in n_C(\alpha)} \left[\sum_{j \in n_I(\beta)} \frac{1}{2} \Phi(r_{\beta j}^h) + \mathcal{F}(\rho_{\beta}^h) \right] \quad (2.24)$$

with weights given by $w_{\alpha} = w_{\alpha,\text{tess.}}/|n_C(\alpha)|$, i.e., by the weight obtained from tessellation of all $N_{s,\text{Cl}}$ cluster center atoms divided by the number of atoms within each cluster. Here, consistency to first order requires that

$$\sum_{i=1}^N E_i = \sum_{\alpha=1}^{N_{s,\text{Cl}}} w_{\alpha} \sum_{\beta \in n_C(\alpha)} E_{\beta} \quad (2.25)$$

$$= E_0 \left(\sum_{\alpha=1}^{N_{s,\text{Cl}}} w_{\alpha} \right) + \mathbf{E}_1 \cdot \left(\sum_{\alpha=1}^{N_{s,\text{Cl}}} w_{\alpha} \sum_{\beta \in n_C(\alpha)} \mathbf{X}_{\beta} \right) \quad (2.26)$$

$$= N \left(E_0 + \mathbf{E}_1 \cdot \sum_{\alpha=1}^{N_{s,\text{Cl}}} w_{\alpha} \mathbf{X}_{s,\alpha,\text{Cl}} \right). \quad (2.27)$$

Comparison with (2.21) shows that the chosen scheme is first-order consistent only if every cluster center \mathbf{X}_{α} is located at the geometric average of all atoms contained within its cluster, i.e., if

$$\mathbf{X}_{s,\alpha} = \frac{1}{N} \sum_{\beta \in n_C(\alpha)} \mathbf{X}_{\beta}. \quad (2.28)$$

In centrosymmetric lattices, this can generally be assumed to hold, and fails only near full atomistic resolution, where the full set of lattice sites must be partitioned into clusters which are overlapping, and clusters may not remain symmetric. Knap and Ortiz [2001] computed cluster weights by enforcing zeroth- and first-order consistency explicitly (or an approximate version thereof via lumping), which, however, we do not consider here due to the enormous computational expenses of solving a linear system for all weights. Using the above Voronoi-tessellated weights, we thus ensure consistency for all summation rules except those based on clusters for which consistency is achieved only in an approximate sense.

2.3 Analysis of summation rules

In the following analysis, we will investigate the accuracy of several (old and new) energy-based summation rules to assess their accuracy with respect to three crucial aspects:

- (i) What is the *approximation power* of the summation rule, i.e., how well does it approximate the total energy?
- (ii) To what extent do *force artifacts* arise from the summation rule in non-uniform meshes?
- (iii) Does the summation rule offer *seamless bridging* to full atomistics?

Of course, points (i) and (ii) are not independent: force artifacts arise when the total energy is not computed correctly (i.e., when the total energy can seemingly be reduced below that of a uniform lattice which implies non-zero forces in an undeformed lattice with non-uniform repatom distribution). (iii) essentially requires that all repatoms are included in the set of sampling atoms to avoid ad-hoc assumptions.

Unfortunately, to date all error analyses in this context have been limited by one or more of the following: (i) one-dimensional or at most two-dimensional scenarios, (ii) small numbers of repatoms, (iii) simplistic interatomic potentials, and (iv) only uniform repatom spacings or individual mesh interfaces were considered instead of complex geometries with aggressive coarse-graining and non-regular repatom distribution. Therefore, our goal is a general assessment of the accuracy of the various energy-based summation rules with realistic potentials in severely non-uniform meshes containing large numbers of lattice sites in two and three dimensions.

2.3.1 Analysis of summation rules in one dimension

In avoidance of the geometric complexity of higher dimensions, [Eidel and Stukowski \[2009\]](#) and [Iyer and Gavini \[2011\]](#) compared the accuracy of selected energy- and force-based summation rules by studying a representative one-dimensional example. Here, we will follow a similar approach and investigate the approximation power and force artifacts of various summation rules of type $(n_{\text{Cl}}, n_{\text{Q}})$ in one dimension, as shown in [Figure 2.7](#). In particular, we include quadrature summation rules which were not included previously. Of course, the one-dimensional example will not accurately capture the physical reality, yet it allows us to draw important conclusions in a clean and instructive manner, which will lead to improved summation rules in higher dimensions.

2.3.1.1 Energy approximation error

Consider the one-dimensional chain of atoms schematically shown in [Figure 2.8a](#), which consists of two semi-infinite atomic chains with uniform atomic spacing l and repatom spacings $n_1 l$ and $n_2 l$ to

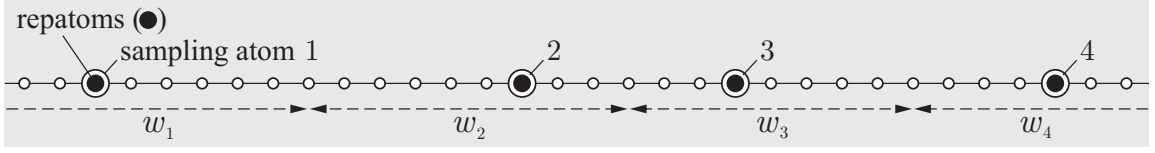
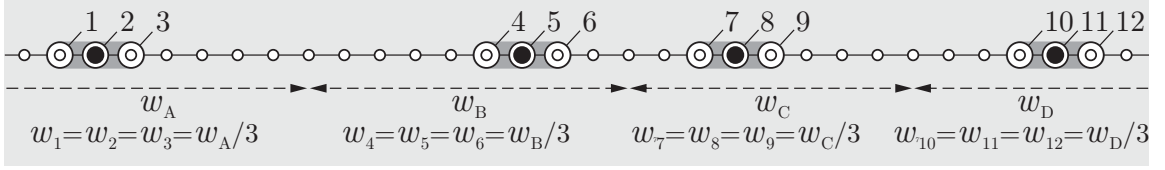
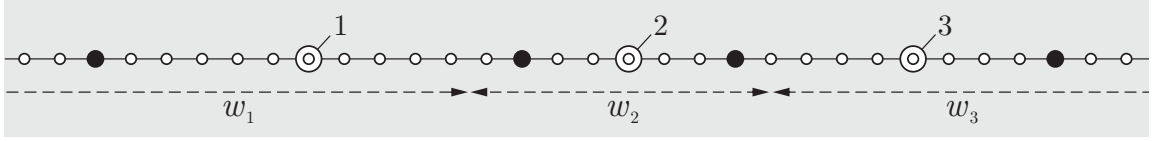
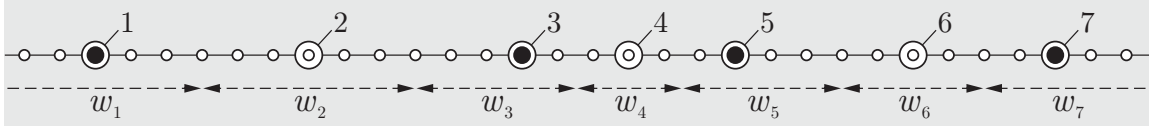
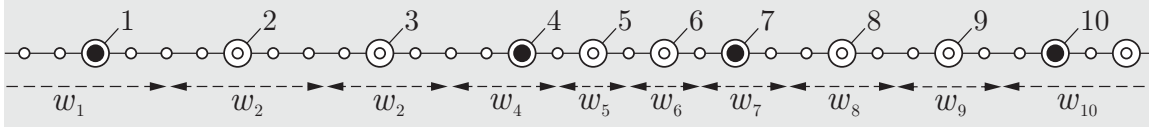
a) nodal summation rule: (0,0)**b) nodal cluster summation rule: (1,0)****c) element quadrature summation rule: (-,1)****d) quadrature summation rule: (0,1)****e) quadrature summation rule: (0,2)**

Figure 2.7: Overview of summation rules in one dimension. Small open circles represent lattice sites, closed circles denote repatoms, and large open circles denote sampling atoms. Weights w_i (one for each sampling atom) are obtained from tessellation.

the left and right of an interface repatom, respectively. Both sides of the interface are assumed to undergo affine deformations changing atomic spacings into $F_1 l$ and $F_2 l$. The radius of interaction of the atomic potential is assumed to involve n_b neighbors. We assume that both elements are sufficiently large, i.e., $n_1, n_2 \gg n_b$. Because linear interpolation results in affine deformation on either side of the interface, atoms located at distances greater than or equal to n_b from the interface have the same energy E_1 (to the left of the interface) or E_2 (to its right), see Figure 2.8b. The energy of the $(2n_b - 1)$ atoms near the interface deviates from these two values; we denote their average energy per atom by \bar{E}_0 . Therefore, the two elements adjacent to the interface have the exact total energy (counting repatoms as belonging half to each adjacent element)

$$V_{\text{exact}} = (n_1 - n_b + \frac{1}{2}) E_1 + (n_2 - n_b + \frac{1}{2}) E_2 + (2n_b - 1) \bar{E}_0. \quad (2.29)$$

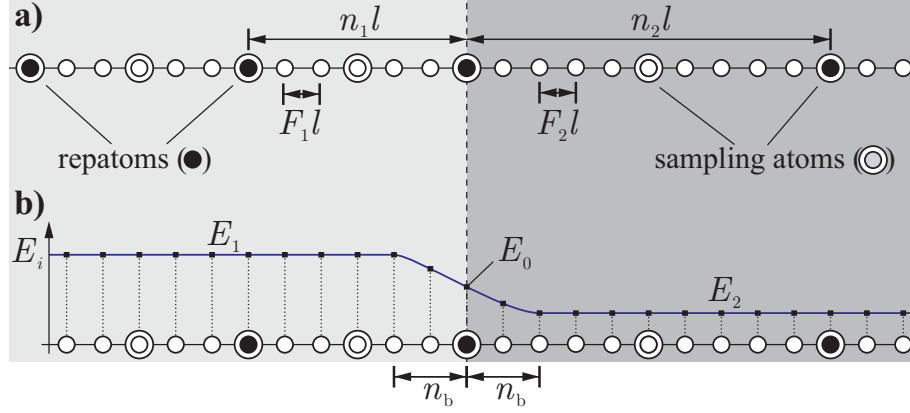


Figure 2.8: a) Example of an interface in a one-dimensional chain of atoms: two semi-infinite atomic chains with repatom spacings $n_1 l$ and $n_2 l$ to the left and right of the interface repatom, respectively. Atoms to the left and right of the interface (originally at equal distances l) experience uniform deformation gradients F_i resulting in atomic spacings $F_1 l$ and $F_2 l$, respectively. b) Schematic illustration of the possible energy per atom along the one-dimensional chain. We note that, here and in the following, the exact form of the energy in the transition between elements depends essentially on the potential and the deformation gradients inside the elements; the shown close-to-linear transition from E_1 to E_2 is only schematic (observable, for example, in a 1D Lennard-Jones crystal with $F_1 = 0.95$, $F_2 = 1.08$ and fourth-nearest neighbor interactions).

nodal summation rule: In the summation rule (0,0) of Figure 2.7a sampling atoms coincide with repatoms, and sampling atom weights to the left and to the right the interface are n_1 or n_2 , respectively. The interface repatom has weight $(n_1 + n_2)/2$. The interface lattice site is assumed to have energy E_0 which may deviate from the average of its neighbors, \bar{E}_0 . The approximate total energy of the two interface elements is now given by

$$\tilde{V}_{(0,0)} = \frac{n_1}{2} E_1 + \frac{n_2}{2} E_2 + \frac{n_1 + n_2}{2} E_0. \quad (2.30)$$

Only in the unstrained ground state (i.e., when $F_1 = F_2 = 1$ and therefore $E_1 = E_2 = E_0 = \bar{E}_0$) the approximate energy becomes exact, i.e. $\tilde{V} = V_{\text{exact}}$. In the limit of large elements ($n_1, n_2 \rightarrow \infty$), the nodal rule leads to the relative error

$$\left| \frac{\tilde{V}_{(0,0)} - V_{\text{exact}}}{V_{\text{exact}}} \right| = \left| \frac{1}{2} \frac{(E_0 - E_1)n_1 + (E_0 - E_2)n_2 - (2\bar{E}_0 - E_1 - E_2)(2n_b - 1)}{(n_1 - n_b + \frac{1}{2})E_1 + (n_2 - n_b + \frac{1}{2})E_2 + (2n_b - 1)\bar{E}_0} \right| \quad (2.31)$$

$$\rightarrow \left| \frac{1}{2} \frac{(E_1 - E_0) + (E_2 - E_0) \frac{n_2}{n_1}}{E_1 + E_2 \frac{n_2}{n_1}} \right| = O(1), \quad (2.32)$$

a deficiency of this type of summation rule that was already pointed out by Iyer and Gavini [2011]. The error persists in uniform meshes (i.e., if $n_1 = n_2$) and only vanishes if deformation gradients are uniform ($F_1 = F_2$). Physically, this large magnitude of error can be expected since the energies of inner-element lattice sites are misrepresented by the energy of the interface sampling atom.

nodal cluster summation rule: For the summation rule $(n_{C1}, 0)$ shown in Figure 2.7b, the above derivation equally applies except that E_0 must be replaced by the energy average of all atoms in the cluster around the interface repatom. Therefore, $|E_0 - \bar{E}_0|$ decreases with increasing cluster radius and vanishes if $n_c \geq n_b$. However, the above scaling for large elements still applies so that the approximation error of all node-based summation rules in the large-element limit is significant.

element quadrature rule: Next, consider an *element quadrature rule* of lowest order $(-, 1)$ as shown in Figure 2.7c. Sampling atoms are located at element centers and therefore have weights n_1 or n_2 . The approximate total energy of the two elements simply becomes

$$\tilde{V}_{(-,1)} = n_1 E_1 + n_2 E_2 \quad (2.33)$$

and the relative energy error assumes the form (with abbreviation $n_i \sim n_1, n_2$)

$$\left| \frac{\tilde{V}_{(-,1)} - V_{\text{exact}}}{V_{\text{exact}}} \right| = \left| \frac{(2n_b - 1) \left[\frac{1}{2}(E_1 + E_2) - \bar{E}_0 \right]}{\left(n_1 - n_b + \frac{1}{2} \right) E_1 + \left(n_2 - n_b + \frac{1}{2} \right) E_2 + (2n_b - 1) \bar{E}_0} \right| = O\left(\frac{1}{n_i}\right). \quad (2.34)$$

As the element size increases ($n_1, n_2 \gg n_b$), quadrature summation rules hence approximate the total energy of the coarse-grained ensemble with increasing accuracy. This result, of course, also holds true for higher-order quadrature rules $(-, n_Q)$ as well: as long as elements are sufficiently large, increasing the number of quadrature atoms per element, n_Q , does not alter the approximated energy since all inner-element quadrature atoms will contribute the same energy. This effect is the aforementioned reason why we do not consider the case of clusters around quadrature sampling atoms, which will be described shortly.

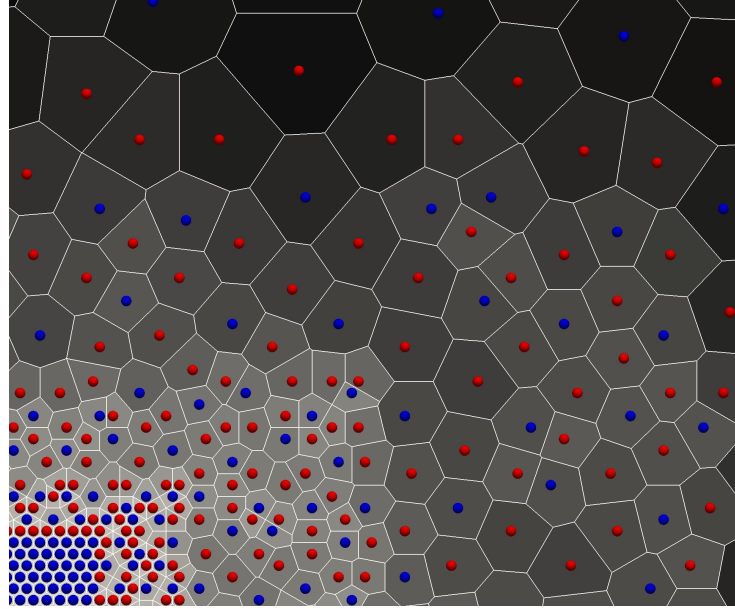
quadrature rule: Finally, the summation rule $(0, n_Q)$ (as shown in Figure 2.7d-e) combines element quadrature and nodal summation rules, i.e., the set of sampling atoms includes all repatoms as well as n_Q equally-spaced quadrature-type sampling atoms per element. In this case, all sampling atoms to the right and to the left of the interface have weights $n_i/(n_Q + 1)$ ($i = 1, 2$, respectively) and the interface sampling atom has weight $(n_1 + n_2)/(2(n_Q + 1))$. The approximate total energy of the two interface-adjacent elements follows as

$$\tilde{V}_{(0,n_Q)} = \frac{2n_Q + 1}{2(n_Q + 1)}(n_1 E_1 + n_2 E_2) + \frac{n_1 + n_2}{2(n_Q + 1)} E_0 \quad (2.35)$$

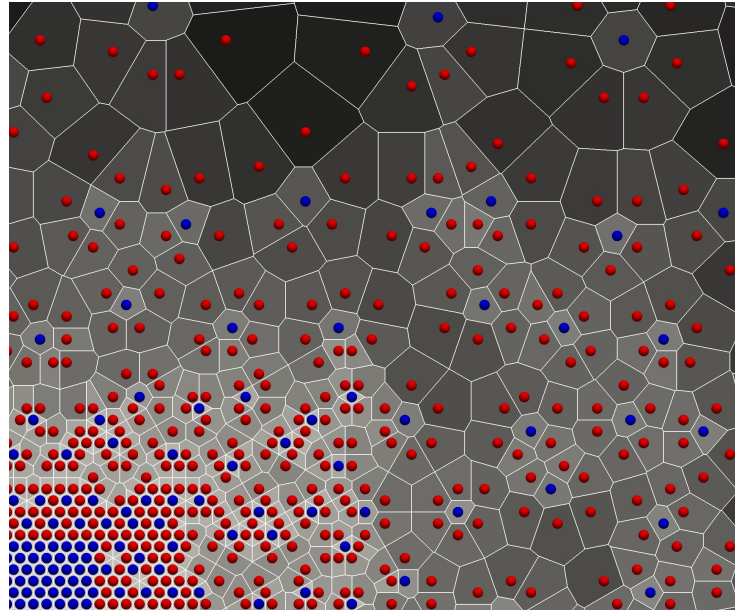
and the large-element limit shows that the relative energy error is

$$\left| \frac{\tilde{V}_{(0,n_Q)} - V_{\text{exact}}}{V_{\text{exact}}} \right| \rightarrow \frac{1}{n_Q + 1} \left| \frac{\tilde{V}_{(0,0)} - V_{\text{exact}}}{V_{\text{exact}}} \right| = O(1). \quad (2.36)$$

Consequently, for quadrature summation rules all qualitative conclusions drawn above for the nodal summation rule apply with the total error reduced by a factor of $n_Q + 1$.



(a) summation rule (0,1)



(b) summation rule (0,3)

Figure 2.9: Tessellated weights for quadrature summation rules.

Regarding clusters around quadrature sampling atoms: Provided that the elements are large enough such that the quadrature sampling atoms do not interact with boundaries of the

elements, all quadrature sampling atoms will compute the same energy and it seems superfluous to use more than one. However, even though the energies of different quadrature sampling atoms are the same, because the weights are calculated via tessellation, using more quadrature atoms can improve the approximation of the energy because more weight is placed on the inner-element energies than on the nodal energies. This can be seen in Figure 2.9, where the areas associated with the repatoms (blue spheres) are smaller with more quadrature points, but still increasing with element size (from bottom left to top right). This observation is the basis for the optimal summation rule described below. Similarly to added quadrature sampling atoms, any cluster sampling atoms around quadrature atoms will also compute the same energy. However, as described in Section 2.2.1, adding cluster sampling atoms around quadrature atoms does not increase the weight placed on the inner-element energies; it only changes the inner-element energy by averaging around a cluster rather than choosing a single point. Due to the fact that the energy of all sampling atoms in the interior of elements are the same, averaging does not improve the accuracy of the inner-element weight. Because of this, summation rules that employ cluster sampling atoms around quadrature sampling atoms are not considered.

Summary: It is important to note that all of the aforementioned summation rules produce the exact energy if the entire crystal is affinely deformed so that $F_1 = F_2$ and hence $E_1 = E_2 = E_0 = \bar{E}_0$. This can be expected since the summation rules are zeroth-order consistent and therefore reproduce a constant field exactly. In summary, the only summation rule with vanishing energy approximation error in large elements is based on element quadrature which, however, we exclude from our fully-nonlocal QC investigation for the following reasons.

The envisioned nonlocal QC formulation does not methodologically differentiate between fully-atomistic and coarse-grained regions and therefore can adaptively adjust the mesh resolution locally in a fully-automatic fashion. There is no notion of an interface between the two regions, and the transition is seamless. Of course, the local/nonlocal QC framework can also bridge to full atomistic resolution by refining elements down to the terminal atomistic level; see for example [Prudhomme et al., 2006, Ortner and Shapeev, 2013]. However, repatoms in local/nonlocal QC are treated fundamentally differently in atomistic or coarse-grained regions (for example, the construction of deformed neighborhoods and the calculation of repatom forces follow different algorithms in those two regions). The presented fully-nonlocal formulation does not differentiate between local and nonlocal elements, and it does not switch from an element-based formulation to a node-based formulation in the atomistic limit – in fact, our QC implementation applies the same algorithms to all repatoms. For all these reasons, we exclude element-based quadrature rules in the following.

2.3.1.2 Residual forces

$$n_1 F_1 + n_2 F_2 = n_1 + n_2 \quad \Rightarrow \quad dF_2 = -\frac{n_1}{n_2} dF_1. \quad (2.37)$$
$$\tilde{F}_0 = -\frac{\partial \tilde{V}}{\partial x_0}(F_1, F_2) = -\frac{\partial \tilde{V}}{\partial F_1} \frac{\partial F_1}{\partial x_0} - \frac{\partial \tilde{V}}{\partial F_2} \frac{\partial F_2}{\partial x_0} = -\left(\frac{\partial \tilde{V}}{\partial F_1} - \frac{n_1}{n_2} \frac{\partial \tilde{V}}{\partial F_2}\right) \frac{\partial F_1}{\partial x_0} \quad (2.38)$$
$$\tilde{F}_0 = -\frac{1}{n_1 l} \left[\frac{\partial \tilde{V}}{\partial E_1} \frac{\partial E_1}{\partial F_1} - \frac{n_1}{n_2} \frac{\partial \tilde{V}}{\partial E_2} \frac{\partial E_2}{\partial F_2} + \frac{\partial \tilde{V}}{\partial E_0} \left(\frac{\partial E_0}{\partial F_1} - \frac{n_1}{n_2} \frac{\partial E_0}{\partial F_2} \right) + \frac{\partial \tilde{V}}{\partial E_L} \frac{\partial E_L}{\partial F_1} - \frac{n_1}{n_2} \frac{\partial \tilde{V}}{\partial E_R} \frac{\partial E_R}{\partial F_2} \right], \quad (2.39)$$

When considering forces on reptatoms in an affinely-strained ground state (with initially uniform lattice deformation gradient F) and elements being sufficiently large ($n_1, n_2 \geq 2n_b$) so that E_1 and

E_2 arise from uniform straining by F_1 and F_2 , respectively, we have

$$\left. \frac{\partial E_1}{\partial F_1} \right|_{F_1=F} = \left. \frac{\partial E_2}{\partial F_2} \right|_{F_2=F}. \quad (2.40)$$

Because of symmetry in the one-dimensional setting and, again, if elements are sufficiently large, we also have

$$\left. \frac{\partial E_0}{\partial F_1} \right|_{F_1=F} = \left. \frac{\partial E_0}{\partial F_2} \right|_{F_2=F} = \left. \frac{\partial E_R}{\partial F_1} \right|_{F_1=F} = \left. \frac{\partial E_L}{\partial F_2} \right|_{F_2=F}. \quad (2.41)$$

Furthermore, note that $\partial V / \partial E_0 = w_0$ is the weight of the sampling atom at the interface and $\partial V / \partial E_i = w_{i,\text{total}}$ is the sum of all weights of sampling atoms in element i (not including sampling atoms on repatom positions). w_L and w_R are the weights of the repatoms to the left and to the right of the interface, respectively. Overall we thus arrive at the residual force on the interface repatom in an affinely-strained ground state:

$$\begin{aligned} \tilde{F}_0 = & - \left[\frac{w_{1,\text{total}}}{n_1} - \frac{w_{2,\text{total}}}{n_2} \right] \frac{1}{l} \left. \frac{\partial E_1}{\partial F_1} \right|_{F_1=F} + w_0 \frac{n_1 - n_2}{n_1 n_2} \frac{1}{l} \left. \frac{\partial E_0}{\partial F_1} \right|_{F_1=F} \\ & + \left[\frac{w_R}{n_2} - \frac{w_L}{n_1} \right] \frac{1}{l} \left. \frac{\partial E_L}{\partial F_1} \right|_{F_1=F}. \end{aligned} \quad (2.42)$$

For an interatomic potential of the Embedded Atom Method [Daw and Baskes, 1984] we can further specify

$$\begin{aligned} \left. \frac{\partial E_1}{\partial F_1} \right|_{F_1=F} &= \left. \frac{\partial}{\partial F_1} \left[\sum_{i=1}^{n_b} \Phi(F_1 i l) + \mathcal{F} \left(2 \sum_{i=1}^{n_b} \phi(F_1 i l) \right) \right] \right|_{F_1=F} \\ &= l \sum_{i=1}^{n_b} i \left[\Phi'(F i l) + 2 \mathcal{F}'(\rho_0) \phi'(F i l) \right], \end{aligned} \quad (2.43)$$

where

$$\rho_0 = 2 \sum_{i=1}^{n_b} \phi(F i l) \quad (2.44)$$

is the effective electron density of an atom in an atomic chain uniformly deformed by F . Analogously, we have

$$\begin{aligned} \left. \frac{\partial E_0}{\partial F_1} \right|_{F_1=F} &= \left. \frac{\partial E_L}{\partial F_1} \right|_{F_1=F} \\ &= \left. \frac{\partial}{\partial F_1} \left[\sum_{i=1}^{n_b} \frac{1}{2} \Phi(F_1 i l) + \sum_{i=1}^{n_b} \frac{1}{2} \Phi(F_1 i l) + \mathcal{F} \left(\sum_{i=1}^{n_b} \phi(F_1 i l) + \sum_{i=1}^{n_b} \phi(F_1 i l) \right) \right] \right|_{F_1=F} \\ &= \frac{1}{2} \left. \frac{\partial E_1}{\partial F_1} \right|_{F_1=F}. \end{aligned}$$

Altogether, this results in

$$\tilde{F}_0 = \frac{n_1 (w_{2,\text{total}} + \frac{1}{2}w_0 + \frac{1}{2}w_R) - n_2 (w_{1,\text{total}} + \frac{1}{2}w_0 + \frac{1}{2}w_L)}{n_1 n_2} \frac{1}{l} \frac{\partial E_1}{\partial F_1} \Big|_{F_1=F}. \quad (2.45)$$

The undeformed ground state of the crystal lattice is characterized by an energy minimum with respect to the atomic spacing, so that (2.43) and thus (2.45) must vanish in the undeformed lattice, i.e., when $F = \mathbf{I}$. Therefore, in a one-dimensional non-uniform QC chain with affine interpolation and sufficiently-large elements none of the summation rules will produce residual forces. In reality, however, marginal numerical errors of the lattice spacing are sufficient to render (2.45) non-zero and thus to produce residual forces. If the chain undergoes uniform affine deformation with $F_1 = F_2 = F$, all lattice sites (in the exact atomistic scenario) should still be force-free but (2.45) will produce forces on repatoms, a clear indication of spurious forces.

Let us evaluate these spurious forces *in the large element limit* for the above summation rules applied to an affine deformation with gradient $F \neq \mathbf{I}$.

nodal summation rule: For the rule (0,0), there are no quadrature-type sampling atoms within elements ($w_{1,\text{total}} = w_{2,\text{total}} = 0$) and the weights of sampling atoms at repatom locations are given by $w_L = n_1$, $w_R = n_2$, and $w_0 = (n_1 + n_2)/2$. Insertion into (2.45) yields the spurious force on the interface repatom arising from the nodal summation rule:

$$\tilde{F}_0^{(0,0)} = \frac{n_1^2 - n_2^2}{4 n_1 n_2} \frac{1}{l} \frac{\partial E_1}{\partial F_1} \Big|_{F_1=F}. \quad (2.46)$$

Apparently, the spurious force at the mesh interface only disappears if the mesh is uniform ($n_1 = n_2$). The addition of clusters containing n_c lattice sites around repatoms does not change this conclusion as long as elements are sufficiently large (i.e., if $n_1, n_2 \geq 2n_b$ and $n_1, n_2 \geq n_c$).

Eidel and Stukowski [2009] presented the one-dimensional example of a chain of two periodically-repeated elements of different sizes as shown in Figure 2.11. For small element sizes, including those analyzed in [Eidel and Stukowski, 2009], the energy obtained from a nodal summation rule is no longer minimal in the undeformed ground state but the crystal can seemingly reduce its energy by displacing the central repatom by u . Figure 2.11 illustrates the energy vs. u for various element sizes in the undeformed ground state using a Lennard-Jones potential (with $n_b = 4$). Here we point out that, for element sizes $n_1, n_2 \geq 2$, the energy landscape becomes symmetric and all residual forces vanish.

element quadrature rules: The rules $(-, n_Q)$ choose sampling atoms within elements and do not include repatoms in the set of sampling atoms, i.e., we have $w_0 = w_L = w_R = 0$ and $w_{i,\text{total}} = n_i$.

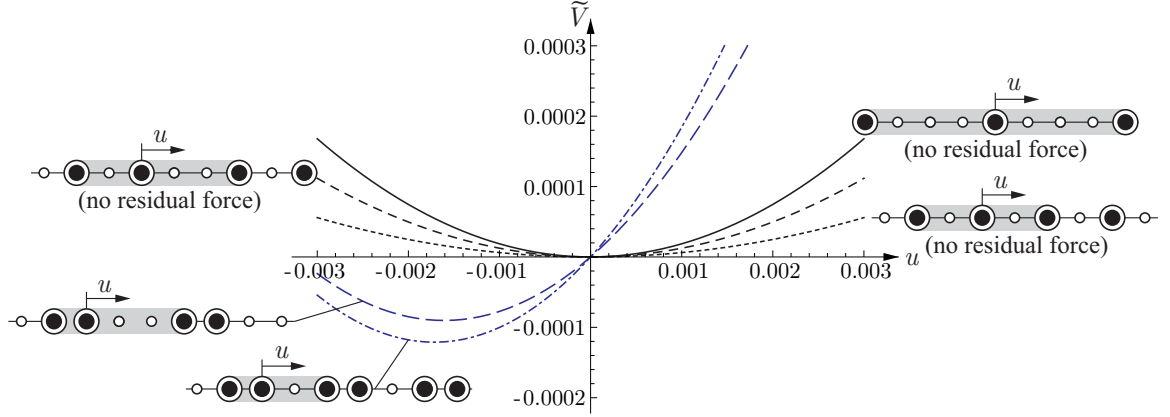


Figure 2.11: Approximated energy of a periodic one-dimensional QC chain described by a Lennard-Jones potential vs. the displacement u of the central repatom for five different discretizations (see schematics) using the nodal summation rule $(0,0)$. Non-uniform meshes only show residual forces if elements are sufficiently small (for example, see the two bottom left examples whose energy minimum is not at $u = 0$ compared to all other examples which do not exhibit a residual force).

Therefore, the residual force (2.45) is reduced to

$$\tilde{F}_0^{(-,n_Q)} = \frac{n_1 n_2 - n_2 n_1}{n_1 n_2} \frac{1}{l} \frac{\partial E_1}{\partial F_1} \Big|_{F_1=F} = 0. \quad (2.47)$$

Consequently, element quadrature rules do not produce residual forces in large elements for any choice of F . This is exactly why the local QC region of Tadmor et al. [1996] does not produce any residual forces (if elements are sufficiently large, it does not make a difference if one uses the Cauchy-Born rule or computes the energy of an actual inner-element lattice site based on its affinely-deforming local neighborhood).

Finally, *quadrature summation rules* of type $(0, n_Q)$ lead to weights

$$w_L = \frac{n_1}{n_Q + 1}, \quad w_R = \frac{n_2}{n_Q + 1}, \quad w_{i,\text{total}} = n_i \frac{n_Q}{n_Q + 1}, \quad w_0 = \frac{w_L + w_R}{2} = \frac{n_1 + n_2}{2(n_Q + 1)}, \quad (2.48)$$

which in turn result in the following residual force at the interface:

$$\tilde{F}_0^{(0,n_Q)} = \frac{n_1^2 - n_2^2}{4 n_1 n_2 (n_Q + 1)} \frac{1}{l} \frac{\partial E_1}{\partial F_1} \Big|_{F_1=F} = \frac{1}{n_Q + 1} \tilde{F}_0^{(0,0)}(x_0). \quad (2.49)$$

Like the energy approximation error, the residual force is reduced by the addition of quadrature sampling atoms, yet it only vanishes in uniform meshes with $n_1 = n_2$ or in the undeformed ground state ($F = F_0$).

2.3.1.3 Optimal summation rule

Taking insight from the above observations, let us conceive a summation rule which does not exhibit spurious force artifacts in any affinely-deformed non-uniform QC mesh ($n_1 \neq n_2$) in one dimension. To allow for seamless bridging to full atomistics (and to avoid computational expenses of clusters), we assume a quadrature rule $(0, x)$ whose set of sampling atoms includes all repatoms and x additional inner-element sampling atoms. According to (2.45), vanishing forces require

$$w_{2,\text{total}} + \frac{1}{2}w_0 + \frac{1}{2}w_R = n_2 \quad \text{and} \quad w_{1,\text{total}} + \frac{1}{2}w_0 + \frac{1}{2}w_L = n_1. \quad (2.50)$$

From all possible solutions, the simplest is to assign the same weight w_{rep} to each sampling atom at a repatom location. To maintain zeroth-order consistency, we choose the remaining sampling atom weights accordingly. This results in

$$w_0 = w_L = w_R = w_{\text{rep}}, \quad w_{i,\text{total}} = n_i - w_{\text{rep}}. \quad (2.51)$$

The choice (2.51) makes (2.45) vanish. Therefore, this summation rule does not produce any residual or spurious forces for arbitrary affine deformation gradients F . We note that the recent summation rule proposed by Beex et al. [2014b] forms a special case of this scenario by defining $w_{\text{rep}} = 1$. Here instead, we identify w_{rep} by minimizing the energy approximation error. In analogy to our previous derivations, we obtain the total energy of the two interface-adjacent elements as

$$\tilde{V} = w_{\text{rep}} E_0 + (n_1 - \tfrac{1}{2}w_{\text{rep}}) E_1 + (n_2 - \tfrac{1}{2}w_{\text{rep}}) E_2 \quad (2.52)$$

so that the relative energy error of the two elements becomes

$$\left| \frac{\tilde{V}_{(0,x^*)} - V_{\text{exact}}}{V_{\text{exact}}} \right| = \left| \frac{(2n_b - 1 - w_{\text{rep}})(E_1 + E_2) + 2(E_0 w_{\text{rep}} - \bar{E}_0(2n_b - 1))}{2V_{\text{exact}}} \right|. \quad (2.53)$$

Hence, the relative energy error in large elements is minimal if

$$w_{\text{rep}} = 2n_b - 1, \quad (2.54)$$

which leads to

$$\left| \frac{\tilde{V}_{(0,x^*)} - V_{\text{exact}}}{V_{\text{exact}}} \right| = (2n_b - 1) \left| \frac{E_0 - \bar{E}_0}{V_{\text{exact}}} \right| = O\left(\frac{n_b}{n_i}\right). \quad (2.55)$$

This relative energy error is bounded and decreases rapidly as elements become larger than the radius of atomic interactions. The error scales with the difference between the interface atom's energy and the average energy of all lattice sites within its radius of interaction. For many EAM-type potentials

(in particular for most pair potentials) we have $E_0 \approx \bar{E}_0$, so that the error is marginal. We note that one can even force the energy error to vanish by replacing sampling atoms on repatoms by clusters of sampling atoms around repatoms containing $2n_b - 1$ lattice sites, in which case we automatically recover $\bar{E}_0 = E_0$.

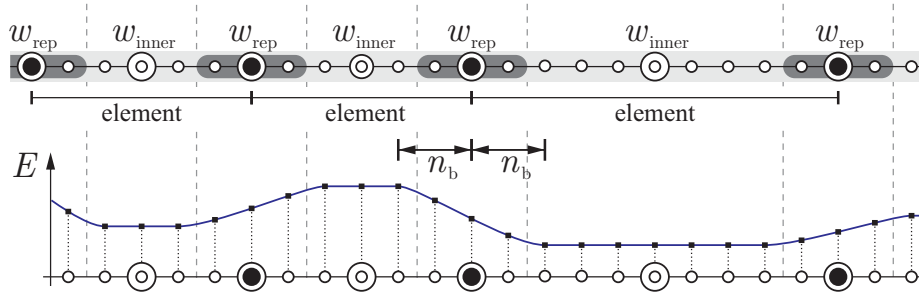


Figure 2.12: Schematic view of the new summation rule: sampling atoms at repatom locations have identical weights w_{rep} , whereas quadrature-type sampling atoms within elements have element-size-dependent weights w_{inner} . The energy is constant within elements and only varies within the radius of interaction of repatoms; all affected lattice sites are represented by the sampling atoms at repatom locations. As in Figure 2.8, the exact form of the energy in the transition between elements depends on the potential and the elements' deformation gradients; the shown close-to-linear transition is only schematic (observable, e.g., in a 1D Lennard-Jones crystal at moderate strains).

Finally, it is important to note that the exact location of the quadrature-type sampling atoms in this new scheme is irrelevant, as long as these are contained well within elements so their neighboring lattice sites undergo an affine deformation of Cauchy-Born type. Also, the number of such sampling atoms is irrelevant (since they all contribute the same energy) and it is sufficient to choose only one such sampling atom. This implies furthermore that we do not even have to specify an actual lattice site for the sampling atom location (as commonly done in quadrature schemes where Gaussian quadrature point-nearest lattice sites are chosen, cf. [Beex et al., 2014b]). One might as well introduce one (fictitious) sampling atom per element which undergoes affine Cauchy-Born deformation and whose weight is defined by $w_{i,\text{total}}$ from (2.51) with (2.54). We will refer to this summation rule in the following as $(0, 1^*)$.

The new summation rule $(0, 1^*)$ allows for a clean physical interpretation as schematically shown in Figure 2.12: each repatom is turned into a sampling atom which represents all lattice sites within the radius of interaction of the repatom (these lattice sites are well represented by the repatom because their interactions reach into the adjacent element). All remaining lattice sites within an element are represented by a single quadrature-type sampling atom which undergoes the same affine deformation as all inner-element lattice sites.

2.3.2 Analysis of summation rules in two dimensions

Inspired by the one-dimensional analysis, we proceed to review summation rules in two dimensions with the ambition to derive a similar new and superior summation rule with vanishing force artifacts and minimal energy approximation error in the limit of large elements.

2.3.2.1 Force artifacts in two dimensions

Let us consider an arbitrarily coarse-grained patch of simplicial (linear, triangular) elements in two dimensions and assume that all elements are sufficiently large such that all neighboring lattice sites within the circle of interaction of any repatom are contained within elements adjacent to that repatom only. Such a scenario is shown in Figure 2.13a, where five triangular elements meet at the central repatom. 2.13a shows a patch of elements with repatoms at element vertices (large solid circles), sampling atoms at vertices and within elements (large open circles), and neighboring lattice sites (small dots) contained within the dashed circles of interaction around each sampling atom. Lattice sites in dark gray regions lie outside the patch; their position remains unaffected by the motion of the central repatom. 2.13b shows the recombination of all neighboring lattice sites within the radii of interaction of those repatoms on the vertices of elements 1 and 2 into equivalent semi-circles of interaction. Suppose a quadrature summation rule of type (0,1) whose set of sampling atoms contains all repatoms as well as one additional sampling atom per element located within the element (whose atomic neighborhood undergoes an affine deformation in the Cauchy-Born sense due to the large element sizes). We assume an atomic lattice in the undeformed ground state so that the neighborhoods of all sampling atoms are identical. The lattice can, in principle, be of any type; the only assumption we will introduce here is that it satisfies centrosymmetry. This includes, for example, all cubic lattices such as bcc and fcc and therefore the vast majority of metals and ceramics. Our goal is to derive the spurious or residual force acting on the central repatom due to the non-uniform QC mesh. The affine interpolation scheme ensures that only the positions of lattice sites in the adjacent elements (shown in light gray) are affected by a displacement \mathbf{u} of the central repatom from its original position \mathbf{X}_0 to the new position $\mathbf{x}_0 = \mathbf{X}_0 + \mathbf{u}$. Figure 2.13a illustrates only one example element patch; in the following, we will generalize our analysis to an arbitrary number of n_e adjacent elements defined by n_e repatoms surrounding the central repatom.

Energy and Force Calculations: The total (approximated) potential energy affected by the position \mathbf{x}_0 of the central repatom is thus given by

$$\tilde{V} = \sum_{e=1}^{n_e} w_{e,\text{inner}} E_{e,\text{inner}} + \sum_{i=1}^{n_e} w_i E_{i,\text{vertex}} + w_0 E_0, \quad (2.56)$$

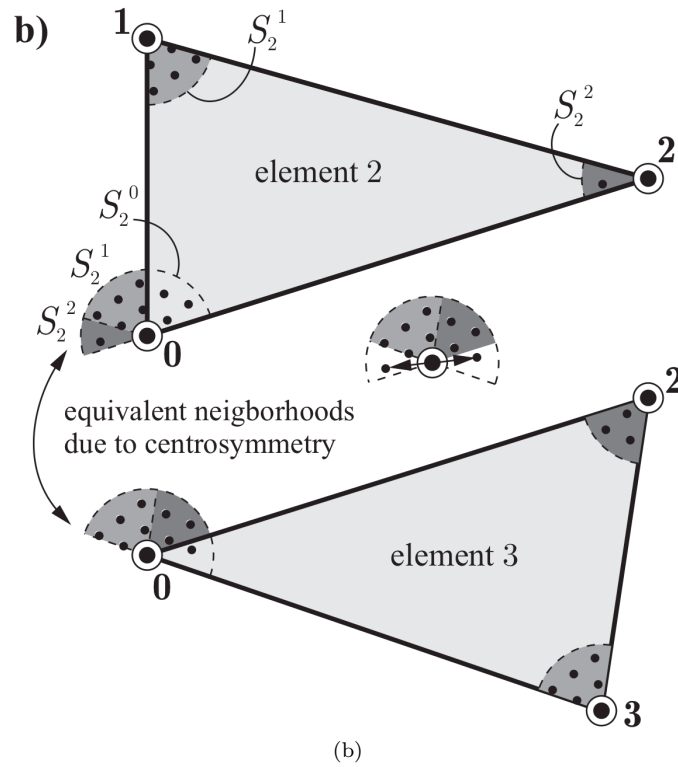
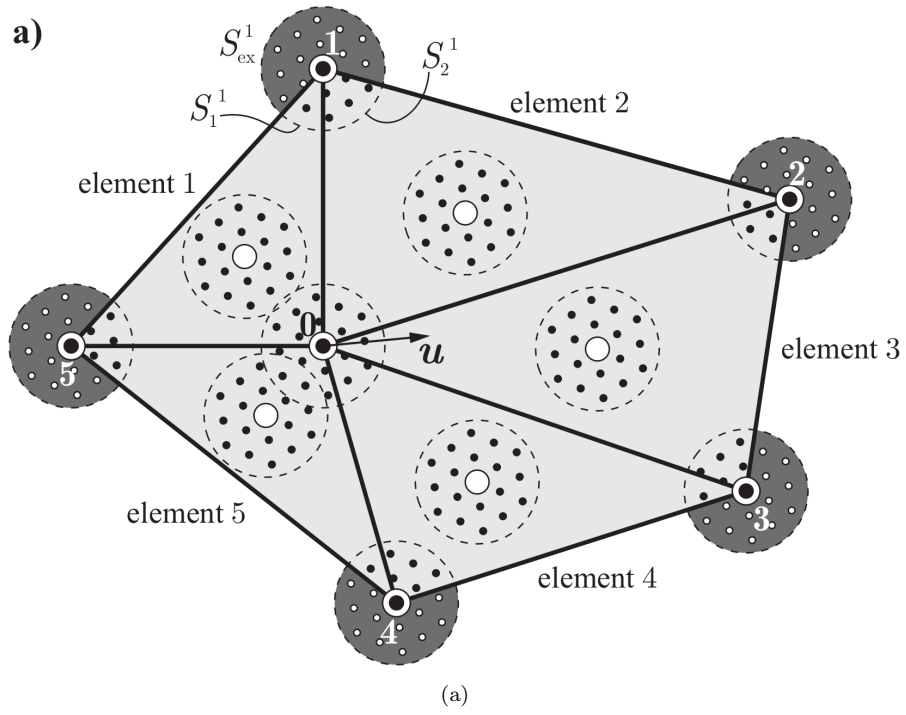


Figure 2.13: Example of a non-uniform patch of triangular elements that result in a residual force on the central repatom 0.

where $E_{e,\text{inner}}$ denotes the energy of the sampling atom located inside element e , $E_{i,\text{vertex}}$ is the energy of outer repatom i , and E_0 is the energy of the central repatom. $w_{e,\text{inner}}$ and w_i denote the weights of those sampling atoms within elements and those on element vertices, respectively. We assume that quadrature-type sampling atoms are chosen sufficiently far away from element edges so that $E_{e,\text{inner}}$ results from Cauchy-Born deformation. The force on the central repatom,

$$\tilde{\mathbf{F}}_0(\mathbf{x}_0) = -\frac{\partial \tilde{V}}{\partial \mathbf{x}_0}(\mathbf{x}_0), \quad (2.57)$$

is determined by differentiation. When using linear interpolation, each element e undergoes an affine deformation characterized by a deformation gradient \mathbf{F}_e (with $F_{iJ} = x_{i,J}$). For concision, let us denote by \mathcal{S} the set of all lattice sites contained within a neighborhood of interaction, and \mathbf{R}_j ($j \in \mathcal{S}$) is the distance vector from the central atom to its neighboring lattice site j . For an EAM potential this leads to

$$E_{e,\text{inner}} = \sum_{j \in \mathcal{S}} \frac{1}{2} \Phi(|\mathbf{F}_e \mathbf{R}_j|) + \mathcal{F}(\rho_e), \quad \rho_e = \sum_{j \in \mathcal{S}} \phi(|\mathbf{F}_e \mathbf{R}_j|). \quad (2.58)$$

Consequently, we obtain

$$\frac{\partial E_{e,\text{inner}}}{\partial \mathbf{x}_0} = \frac{\partial E_{e,\text{inner}}}{\partial \mathbf{F}_e} \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0} = \sum_{j \in \mathcal{S}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_e \mathbf{R}_j|) + \mathcal{F}'(\rho_e) \phi'(|\mathbf{F}_e \mathbf{R}_j|) \right] \frac{\mathbf{F}_e \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_e \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0}. \quad (2.59)$$

For the sampling atoms located on repatom positions on the outer vertices of the patch of elements, the energy calculation is more involved because only those neighboring lattice sites contained within the light-gray adjacent elements depend on the position \mathbf{x}_0 (while those atoms in the dark-gray regions are independent of \mathbf{x}_0). Taking for example the top sampling atom (1) whose neighbors are contained in elements 1 and 2, we have

$$\begin{aligned} E_{1,\text{vertex}} = & \sum_{j \in \mathcal{S}_1^1} \frac{1}{2} \Phi(|\mathbf{F}_1 \mathbf{R}_j|) + \sum_{j \in \mathcal{S}_2^1} \frac{1}{2} \Phi(|\mathbf{F}_2 \mathbf{R}_j|) + \sum_{j \in \mathcal{S}_{\text{ex}}^1} \frac{1}{2} \Phi(|\mathbf{F}_{\text{ex}} \mathbf{R}_j|) \\ & + \mathcal{F} \left[\sum_{j \in \mathcal{S}_1^1} \phi(|\mathbf{F}_1 \mathbf{R}_j|) + \sum_{j \in \mathcal{S}_2^1} \phi(|\mathbf{F}_2 \mathbf{R}_j|) + \sum_{j \in \mathcal{S}_{\text{ex}}^1} \phi(|\mathbf{F}_{\text{ex}} \mathbf{R}_j|) \right], \end{aligned} \quad (2.60)$$

where $\mathcal{S}_e^j = \mathcal{S}^j \cap E_e$ is the set of neighbors of repatom j contained in element e , and $\mathcal{S}_{\text{ex}}^j = \mathcal{S}^j \setminus \bigcup_{e=1}^{n_e} E_e$ are those neighboring lattice sites located outside the patch of elements (E_e denotes the set of all lattice sites in element e). \mathbf{F}_{ex} is the (generally not constant) deformation gradient

acting on the lattice sites not contained in the patch of elements. Consequently, we see that

$$\begin{aligned} \frac{\partial E_{1,\text{vertex}}}{\partial \mathbf{x}_0} &= \frac{\partial E_{1,\text{vertex}}}{\partial \mathbf{F}_1} \cdot \frac{\partial \mathbf{F}_1}{\partial \mathbf{x}_0} + \frac{\partial E_{1,\text{vertex}}}{\partial \mathbf{F}_2} \cdot \frac{\partial \mathbf{F}_2}{\partial \mathbf{x}_0} \\ &= \sum_{e=1}^2 \sum_{j \in \mathcal{S}_e^1} \left[\frac{1}{2} \Phi'(|\mathbf{F}_e \mathbf{R}_j|) + \mathcal{F}'(\rho_1) \phi'(|\mathbf{F}_e \mathbf{R}_j|) \right] \frac{\mathbf{F}_e \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_e \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0} \end{aligned} \quad (2.61)$$

with

$$\rho_1 = \sum_{j \in \mathcal{S}_1^1} \phi(|\mathbf{F}_1 \mathbf{R}_j|) + \sum_{j \in \mathcal{S}_2^1} \phi(|\mathbf{F}_2 \mathbf{R}_j|) + \sum_{j \in \mathcal{S}_{\text{ex}}^1} \phi(|\mathbf{F}_{\text{ex}} \mathbf{R}_j|). \quad (2.62)$$

Analogously, the energies (and their derivatives) of all sampling atoms on the outer vertices can be computed. Finally, the energy of the sampling atom located at the position of the central repatom involves neighboring lattice sites in all adjacent elements. Here we have

$$E_0 = \left[\sum_{e=1}^{n_e} \sum_{j \in \mathcal{S}_e^0} \frac{1}{2} \Phi(|\mathbf{F}_e \mathbf{R}_j|) \right] + \mathcal{F}(\rho_0), \quad \rho_0 = \sum_{e=1}^{n_e} \sum_{j \in \mathcal{S}_e^0} \phi(|\mathbf{F}_e \mathbf{R}_j|) \quad (2.63)$$

and consequently

$$\frac{\partial E_0}{\partial \mathbf{x}_0} = \sum_{e=1}^{n_e} \frac{\partial E_0}{\partial \mathbf{F}_e} \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0} = \sum_{e=1}^{n_e} \sum_{j \in \mathcal{S}_e^0} \left[\frac{1}{2} \Phi'(|\mathbf{F}_e \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_e \mathbf{R}_j|) \right] \frac{\mathbf{F}_e \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_e \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0}. \quad (2.64)$$

The total force acting on the central repatom is now given by inserting (2.59), (2.61), and (2.64) into

$$\tilde{\mathbf{F}}_0(\mathbf{x}_0) = - \left[\sum_{e=1}^{n_e} w_{e,\text{inner}} \frac{\partial E_{e,\text{inner}}}{\partial \mathbf{x}_0} + \sum_{i=1}^{n_e} w_i \frac{\partial E_{i,\text{vertex}}}{\partial \mathbf{x}_0} + w_0 \frac{\partial E_0}{\partial \mathbf{x}_0} \right]. \quad (2.65)$$

Uniformly-strained configuration: Let us determine the spurious/residual force acting on the central repatom in a uniformly strained configuration with deformation gradient \mathbf{F}_0 applied to every element (for example, in the undeformed ground state we have $\mathbf{F}_0 = \mathbf{I}$). Under these conditions, the residual force is given by

$$\tilde{\mathbf{F}}_0 = - \sum_{e=1}^{n_e} \left[\frac{\partial \tilde{V}}{\partial \mathbf{F}_e} \right]_{\mathbf{F}_e = \mathbf{F}_0} \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0}, \quad (2.66)$$

where

$$\frac{\partial \tilde{V}}{\partial \mathbf{F}_e} = \sum_{e=1}^{n_e} w_{e,\text{inner}} \frac{\partial E_{e,\text{inner}}}{\partial \mathbf{F}_e} + \sum_{i=1}^{n_e} w_i \frac{\partial E_{i,\text{vertex}}}{\partial \mathbf{F}_e} + w_0 \frac{\partial E_0}{\partial \mathbf{F}_e}. \quad (2.67)$$

First, consider only those contributions from a particular element e ; take element 2 whose deformation gradient \mathbf{F}_2 is applied to atomic spacings in the circles of interactions of repatoms 0, 1 and 2

and of the sampling atom located inside element 2 (see Figure 2.13). Therefore, we obtain

$$\begin{aligned}
\left[\frac{\partial \tilde{V}}{\partial \mathbf{F}_2} \right]_{\mathbf{F}_2 = \mathbf{F}_0} &= \left[w_{2,\text{inner}} \frac{\partial E_{2,\text{inner}}}{\partial \mathbf{F}_2} + w_1 \frac{\partial E_{1,\text{vertex}}}{\partial \mathbf{F}_2} + w_2 \frac{\partial E_{2,\text{vertex}}}{\partial \mathbf{F}_2} + w_0 \frac{\partial E_0}{\partial \mathbf{F}_2} \right]_{\mathbf{F}_2 = \mathbf{F}_0} \\
&= w_{2,\text{inner}} \sum_{j \in \mathcal{S}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|) \right] \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \\
&\quad + \sum_{\alpha=0}^2 w_\alpha \sum_{j \in \mathcal{S}_2^\alpha} \left[\frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|) \right] \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|}
\end{aligned} \tag{2.68}$$

with

$$\rho_0 = \sum_{j \in \mathcal{S}} \phi(|\mathbf{F}_0 \mathbf{R}_j|). \tag{2.69}$$

The individual segments of the circles of interactions, \mathcal{S}_2^1 , \mathcal{S}_2^2 , and \mathcal{S}_2^0 are shown in Figure 2.13b. For a centrosymmetric lattice, these three segments can be recombined as shown in Figure 2.13b: by translating segments \mathcal{S}_2^1 and \mathcal{S}_2^2 to the central repatom and reflecting \mathcal{S}_2^1 about the latter (due to centrosymmetry), we see that

$$\mathcal{S}_2^1 \cup \mathcal{S}_2^2 \cup \mathcal{S}_2^0 = \mathcal{S}_{1/2}, \tag{2.70}$$

i.e., the three segments recombine to half the circle of interaction denoted by $\mathcal{S}_{1/2}$. How exactly the circle is split in half is irrelevant due to centrosymmetry (see, for example, the two equivalent semi-circles shown in Figure 2.13b) because (2.68) is independent of the sign of all \mathbf{R}_j . In other words, summing over all lattice site neighbors within any semi-circle of interaction will yield the same result in (2.68) as long as all lattice sites within that semi-circle experience the same affine deformation.

Note that the energy of a single atom in a lattice that is affinely deformed by a deformation gradient \mathbf{F}_0 is given by

$$E(\mathbf{F}_0) = \sum_{j \in \mathcal{S}} \frac{1}{2} \Phi(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}(\rho_0), \quad \rho_0 = \sum_{j \in \mathcal{S}} \phi(|\mathbf{F}_0 \mathbf{R}_j|) \tag{2.71}$$

and the first Piola-Kirchhoff stress tensor of the Cauchy-Born continuum follows as [Ericksen, 1984, Tadmor et al., 1996]

$$\mathbf{P}(\mathbf{F}_0) = \frac{\partial E}{\partial \mathbf{F}} \Big|_{\mathbf{F} = \mathbf{F}_0} = \sum_{j \in \mathcal{S}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|) \right] \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|}. \tag{2.72}$$

Here it becomes apparent that summing only over the semi-circle of interaction gives the same result due to centrosymmetry (replacing any of the \mathbf{R}_j by $-\mathbf{R}_j$ in (2.72) does not alter the result), so that

we might as well write

$$\mathbf{P}(\mathbf{F}_0) = 2 \sum_{j \in \mathcal{S}_{1/2}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|) \right] \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|}, \quad (2.73)$$

where, as before, $\mathcal{S}_{1/2}$ denotes any semi-circle of interaction. If the single-crystalline atomic lattice is in its undeformed ground state (i.e., $\mathbf{F}_0 = \mathbf{I}$), we know that the energy is minimized and therefore

$$\mathbf{P}(\mathbf{I}) = \sum_{j \in \mathcal{S}} \left[\frac{1}{2} \Phi'(|\mathbf{R}_j|) + \mathcal{F}'(\rho_0(\mathbf{I})) \phi'(|\mathbf{R}_j|) \right] \frac{\mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{R}_j|} = \mathbf{0}. \quad (2.74)$$

A comparison shows that the term in (2.68) associated with the inner sampling atom (i.e., the term multiplying $w_{2,\text{inner}}$) must vanish in the undeformed ground state. The remaining terms, however, cannot vanish for arbitrary element geometries, unless one chooses $w_1 = w_2 = w_0$ so that the remaining terms recombine into a sum over the semi-circle of interaction.

Observations: In other words, if the weights assigned to sampling atoms at repatom locations depend on the element size and geometry (as is commonly the case in nodal, cluster, or traditional quadrature rules), force artifacts will be unavoidable in non-uniform QC meshes, even in the large-element limit. If the mesh is uniform and sampling atoms are uniformly spaced, these forces vanish due to uniform repatom weights (this is illustrated in 2.3.2.4 for the residual forces at a sharp mesh interface in two dimensions). The observed spurious force artifacts also imply a significant approximation error of the total potential energy due to the summation rules. As shown in one dimension, the local QC formulation based on Cauchy-Born kinematics does not suffer from these forces because it uses $w_i = 0$ for all repatoms, yet it requires to switch from an element-based to a node-based formulation when transitioning to fully-atomistic resolution and to treat atomistic and coarse-grained regions fundamentally differently.

2.3.2.2 Optimal summation rule in two dimensions

Based on the above observations, let us introduce a quadrature summation rule $(0, 1^*)$ with vanishing spurious forces in the large-element limit in two-dimensional non-uniform QC meshes under affine deformation. To this end, we adopt the concept discussed in one dimension: we assign the same weight to all sampling atoms at repatom locations, i.e.,

$$w_0 = w_1 = \dots = w_{n_e} = w_{\text{rep}}. \quad (2.75)$$

Uniformly-strained Configuration: In this case, the force (2.66) on the central repatom simplifies dramatically because the sums over all repatom neighbors recombine into semi-circles of

interaction as discussed above (cf. Figure 2.13b). This is demonstrated in Section 2.3.2.4 for the instructive example of a sharp mesh interface. After insertion of all derivatives analogously to (2.68), (2.66) turns into

$$\tilde{\mathbf{F}}_0 = - \sum_{e=1}^{n_e} \left(w_{e,\text{inner}} + \frac{1}{2} w_{\text{rep}} \right) \left[\sum_{j \in \mathcal{S}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|) \right] \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \right] \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0}. \quad (2.76)$$

By comparison of (2.74) and (2.76), we conclude that in the undeformed ground state ($\mathbf{F}_0 = \mathbf{I}$) no residual forces exist in arbitrary non-uniform meshes (with sufficiently large elements).

Let us proceed to identify sampling atom weights that do not produce spurious forces in any affinely-deformed state with $\mathbf{F}_0 \neq \mathbf{I}$. To this end, we exploit the fact the deformation gradients \mathbf{F}_e within the n_e elements in (2.76) are not independent, since they all result from the central repatom moving to the new location \mathbf{x}_0 while the other repatoms are held fixed. For any simply-connected two-dimensional region Ω with area A_Ω whose outer boundary is deforming affinely according to an imposed deformation gradient \mathbf{F}_0 , the average deformation gradient within Ω must satisfy

$$\begin{aligned} \int_{\Omega} \mathbf{F} \, dA &= \int_{\Omega} \text{Grad} \, \mathbf{x} \, dA = \int_{\partial\Omega} \mathbf{x} \otimes \mathbf{N} \, dA = \int_{\partial\Omega} \mathbf{F}_0 \mathbf{X} \otimes \mathbf{N} \, dA \\ &= \int_{\Omega} \mathbf{F}_0 \, \text{Grad}(\mathbf{X}) \, dA = \int_{\partial\Omega} \mathbf{F}_0 \, dA = \mathbf{F}_0 A_\Omega. \end{aligned} \quad (2.77)$$

For a discrete patch of n_e simplicial elements (as in Figure 2.13) with constant deformation gradients \mathbf{F}_e and areas A_e ($i = 1, \dots, n_e$) the analogous condition becomes

$$\sum_{e=1}^{n_e} A_e \mathbf{F}_e = \mathbf{F}_0 A. \quad (2.78)$$

Its derivation for a general patch of simplicial elements can be found in Section 2.3.2.6, where we account for the fact that the deformation gradient is not continuous across element boundaries so that (2.77) does not strictly apply. An immediate consequence is that

$$\sum_{e=1}^{n_e} A_e \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0} = \mathbf{0}. \quad (2.79)$$

By using (2.79), we can make (2.76) vanish for an arbitrary \mathbf{F}_0 by extending (2.75) into

$$w_0 = w_1 = \dots = w_{n_e} = w_{\text{rep}} \quad \text{and} \quad w_{e,\text{inner}} = A_e - \frac{1}{2} w_{\text{rep}} \quad (2.80)$$

such that

$$\tilde{\mathbf{F}}_0 = - \left[\sum_{j \in \mathcal{S}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|) \right] \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \right] \cdot \sum_{e=1}^{n_e} A_e \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0} = \mathbf{0}. \quad (2.81)$$

It is important to note that, if weights are chosen according to (2.80), not only do residual forces vanish in the undeformed ground state but also in *any other* uniformly strained configuration. Consequently, the new weights (2.80) ensure that an arbitrary non-uniform QC mesh (of sufficiently large elements) produces zero forces on all repatoms under affine deformation. **This is the first (nonlocal) QC summation rule with these features.**

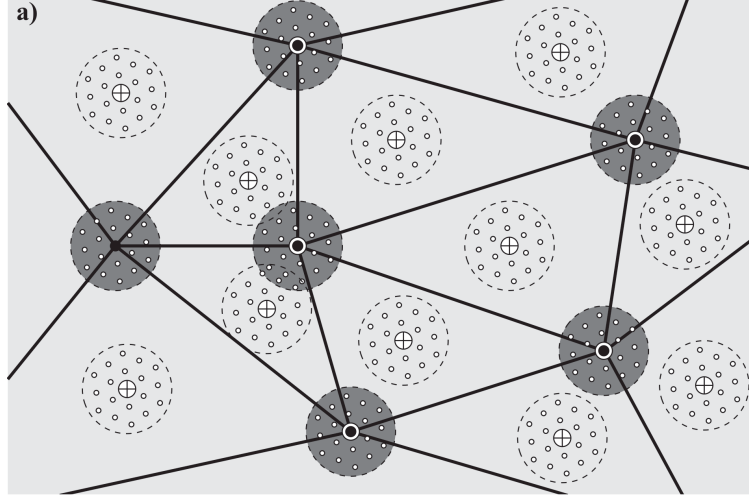


Figure 2.14: Quadrature-type Cauchy-Born neighborhoods in the optimal 2D summation rule.

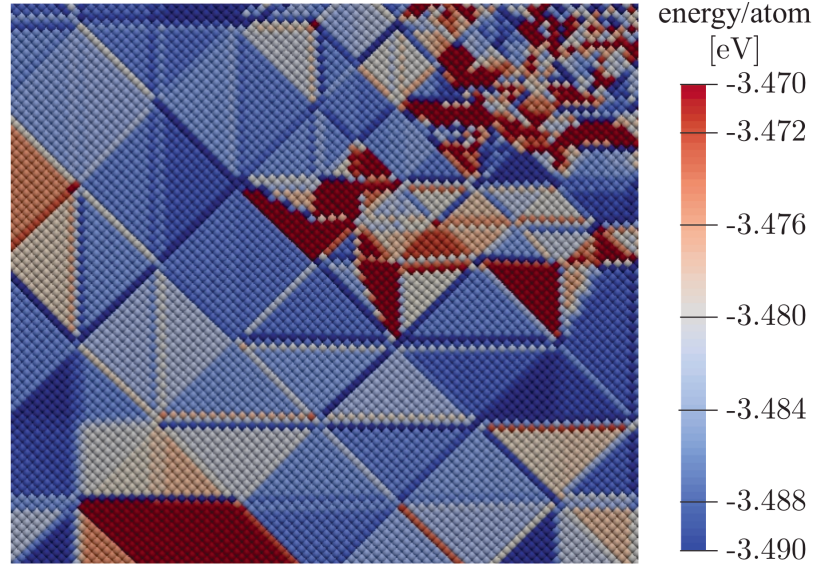
We still have not defined w_{rep} , the weight of individual repatoms. Inspired by the one-dimensional analysis, we aim to minimize the energy approximation error by choosing weights such that each sampling atom at a repatom site represents all those lattice sites within an *effective interaction distance* r_{eff} whose energy is best represented by that of the repatom. The exact value of r_{eff} can be determined by numerical experiments and is reported later for specific examples. Thus we have $w_{\text{rep}} = \pi r_{\text{eff}}^2 \rho$ with atomic density ρ . The sampling atom within each element thus represents all remaining lattice sites not represented by the sampling atoms at nodes. Figure 2.14 shows a schematic illustration of the weights of sampling atoms at repatom sites (dark gray regions) and of quadrature-type sampling atoms for the $(0, 1^*)$ rule applied to a two-dimensional triangular nonuniform QC mesh (quadrature-type sampling atoms need not be actual lattice sites). This completes the $(0, 1^*)$ summation rule in two dimensions.

Physical interpretation: The clean physical interpretation of this summation rule is maintained from the one-dimensional inspiration: each repatom is turned into a sampling atom which represents all lattice sites within a certain effective radius of interaction of the repatom because these lattice sites are fairly well represented by the repatom, and all remaining lattice sites within an element are represented by a single quadrature-type sampling atom which undergoes the same affine deformation as all inner-element lattice sites. This construction and interpretation are further motivated by observing the distribution of energy per lattice site in a QC discretization. Figure 2.15b shows the distribution of the energy per lattice site in a QC discretization with affine interpolation, clearly showing constant energies within elements, variation near the nodes, and stripes of variations near the element boundaries. With this first-order new summation rule, the bulk and nodes are well-represented; a second-order new summation rule that also captures the element boundaries is developed in Section 2.3.2.3.

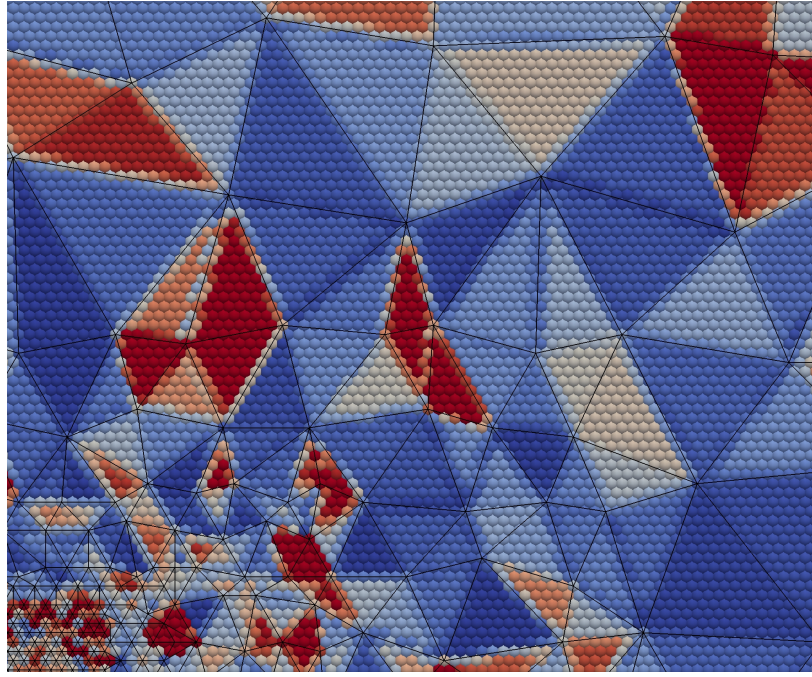
Cauchy-Born: We note that, like in one dimension, the exact location of the inner-element sampling atom is irrelevant as long as its entire neighborhood of interaction is contained within the element. As a matter of fact, we might as well introduce a sampling atom at a fictitious lattice site with a pristine single-crystalline neighborhood deformed by the Cauchy-Born rule according to the element's deformation gradient. Numerically, the Cauchy-Born neighborhood of a fictitious sampling atom can be realized by the same setup by introducing fictitious neighboring lattice sites. In case such neighborhoods reach outside an element, the positions of all neighboring lattice sites are obtained by affine extrapolation (see Appendix 2.3.2.5). Thus, rule $(0, 1^*)$ combines local QC with energy-based nodal QC.

2.3.2.3 Optimal summation rule of second order in two dimensions

The $(0, 1^*)$ summation rule introduced above admits extensions of higher order, which is readily motivated by examining the distribution of energy within elements. Figure 2.15 shows the energy per lattice site in regular (2.15a) and irregular (2.15b) meshes. These plots reveal that within a deformed QC mesh, affine interpolation results in constant energies within elements with deviations near element boundaries. In $(0, 1^*)$, the inner-element sampling atom represents all inner-element lattice sites except those within the effective interaction radius of repatoms. Let us consider the new $(0, 1^*)$ rule as a first-order summation rule (zeroth being the original local QC technique with $(-, 1)$, relying solely on the Cauchy-Born rule for each element). A second-order summation rule follows by adding sampling atom on element edges (see Figure 2.16), one such sampling atom per element edge to represent all those lattice sites on and near the element edges (having semi-spheres of interaction within both elements adjacent to the edge). This second-order scheme will be denoted by $(0, 1^*, 3^*)$ and is shown in Figure 2.18d. In sufficiently large elements, we now choose the following repatom



(a)



(b)

Figure 2.15: Energy per lattice site for regular (top) and irregular (bottom) 2D QC meshes.

weights to minimize energy errors in analogy to the first-order rule $(0, 1^*)$:

$$\begin{aligned}
 w_0 = w_1 = \dots = w_{n_e} = w_{\text{rep}} &= \pi r_{\text{eff}}^2, & w_{i,\text{edge}} &= L_i r_{\text{eff}} \\
 \text{and} \quad w_{e,\text{inner}} &= A_e - \frac{1}{2} w_{\text{rep}} - \sum_{i=1}^3 \frac{w_{i,\text{edge}}}{2}.
 \end{aligned} \tag{2.82}$$

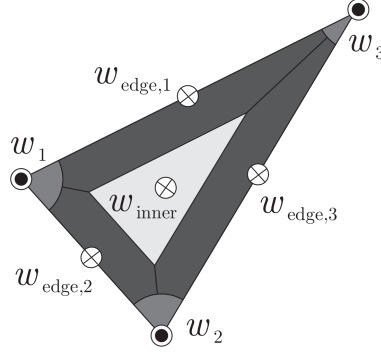


Figure 2.16: Second-order new summation rule.

In addition to our previous definitions, $w_{i,\text{edge}}$ denotes the weight of a sampling atom at an element edge of length L_{edge} . The final sum is over all three element edges of element e . Key to understanding spurious force artifacts is the fact that neighboring lattice sites of each sampling atom on an element edge contain a complete semi-circle of neighbors in either of the two adjacent elements. As explained before, due to centrosymmetry these do not produce residual forces in sufficiently large elements (i.e., when each semi-circle undergoes an affine deformation).

Spurious force artifacts are derived in analogy to the previous case and result in

$$\begin{aligned} \tilde{\mathbf{F}}_0 = & - \sum_{e=1}^{n_e} \left(w_{e,\text{inner}} + \frac{1}{2} w_{\text{rep}} \right) \left[\sum_{j \in \mathcal{S}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|) \right] \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \right] \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0} \\ & - \sum_{e=1}^{n_e} \left(\sum_{i=1}^3 \frac{w_{i,\text{edge}}}{2} \right) \left[\sum_{j \in \mathcal{S}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|) \right] \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \right] \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0}, \end{aligned} \quad (2.83)$$

where the final term contains a sum over all three edges adjacent to element e . If weights are chosen according to (2.82), then (2.83) reduces to (2.81) which vanishes in affinely-deformed meshes. Therefore, the second-order scheme $(0, 1^*, 3^*)$ comes with the same properties as its first-order counterpart: spurious force artifacts vanish in sufficiently large elements for any affine deformation of the non-uniform QC mesh. However, the energy approximation error is considerably reduced by appropriately representing all edge-nearest lattice sites. We note that similar approaches to the second-order summation rule have been introduced in the context of cohesive zone modeling of fracture [Zeng and Li, 2010] as well as in atomistic-to-finite-element coupling techniques [Shan and Nackenhorst, 2010].

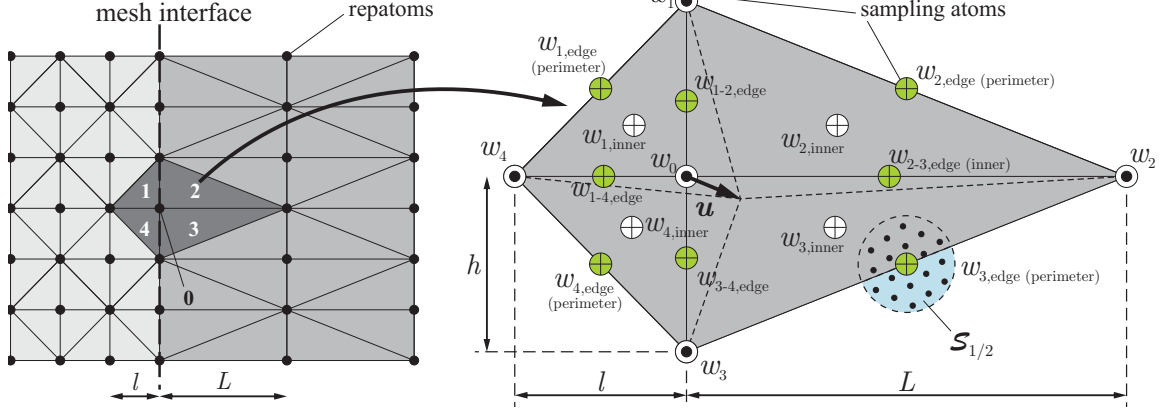


Figure 2.17: Two-dimensional scenario to determine the residual force on a sharp mesh interface with a zoom into the four elements adjacent to repatom 0 at the center. A residual force $\bar{\mathbf{F}}_0$ on repatom 0 emerges if the body can seemingly lower its energy by a displacement \mathbf{u} . Repatoms have a solid filling, sampling atoms have open circles; crosses denote sampling atoms which undergo Cauchy-Born deformation (green sampling atoms on element edges only exist in the new second-order summation rule). The sampling atom on the outer edge of element 3 illustrates a typical atomic neighborhood (the blue semicircle of interaction is denoted by $\mathcal{S}_{1/2}$ in the derivations).

2.3.2.4 Residual force at a sharp mesh interface in two dimensions

Before developing this new summation rule for three dimensions, let us analyze the residual force at sharp mesh interfaces, a common situation in coarse-graining. In general, residual force artifacts arise in non-uniform QC meshes. To illustrate this phenomenon, let us consider a sharp interface in a semi-structured two-dimensional QC mesh with an affine interpolation (as used throughout this paper). Figure 2.17 illustrates such a scenario with elements coarsened in the horizontal direction by a factor of L/l . We will determine the residual force acting on a repatom located at the interface (here labeled 0) by analyzing the patch of its four adjacent elements (shown in dark gray). A residual force on repatom 0 arises if the summation rule furnishes the system with the possibility to reduce its (approximated) total potential energy by displacing the repatom 0. To specialize to the new first- and second-order summation rules of this work, we assume that all repatoms are sampling atoms, and that every element contains one sampling atom in its interior. With appropriate weights, this represents the new first-order rule $(0, 1^*)$. In order to study the new second-order rule $(0, 1^*, 3^*)$ we include additional sampling atoms, viz. one per element edge (as shown in Figure 2.17).

Let us derive the residual force acting on the interface sampling atom 0. The (approximated) total potential energy affected by the position \mathbf{x}_0 of the central repatom is given by the summation over all sampling atoms contained in the four adjacent elements ($e = 1, \dots, 4$), since their atomic

neighborhoods depend on \mathbf{x}_0 through interpolation:

$$\begin{aligned} \tilde{V} = & \sum_{e=1}^4 w_{e,\text{inner}} E_{e,\text{inner}} + w_0 E_0 + \sum_{i=1}^4 w_i E_{i,\text{vertex}} \\ & + \sum_{i=1}^4 w_{i,\text{edge}} E_{i,\text{edge}} + \sum_{e,f} w_{e-f,\text{edge}} E_{e-f,\text{edge}}, \end{aligned} \quad (2.84)$$

where $E_{e,\text{inner}}$ denotes the energy of the sampling atom located inside element e , $E_{i,\text{vertex}}$ is the energy of outer repatom i , E_0 is the energy of the central repatom, and $E_{i,\text{edge}}$ and $E_{e-f,\text{edge}}$ are the energies of sampling atoms located on the outer edge of element i and between elements i and j , respectively (these sampling atoms exist in the second-order summation rule). $w_{e,\text{inner}}$, w_i , w_0 , $w_{e,\text{edge}}$, and $w_{e-f,\text{edge}}$ denote the corresponding weights of the sampling atoms (see Figure 2.17). We assume that that inner-element sampling atoms experiences affine neighborhood deformations of Cauchy-Born type, and that the sampling atoms on edges see affine Cauchy-Born deformation on either side of the element edge. The residual force on repatom 0 thus follows as

$$\tilde{\mathbf{F}}_0 = -\frac{\partial \tilde{V}}{\partial \mathbf{x}_0} = -\sum_{e=1}^4 \frac{\partial \tilde{V}}{\partial \mathbf{F}_e} \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0}. \quad (2.85)$$

For the chosen geometric example, the deformation gradients \mathbf{F}_e in elements 1 through 4 all depend on \mathbf{x}_0 . Assume the central repatom is displaced to the new location $\mathbf{x}_0 = (x_0, y_0) = \mathbf{X} + (u, v)$, whereas the remaining repatoms remain in the undeformed ground state. Since we use an affine interpolation within each element, the deformation gradients in the four elements are constant and given by

$$\begin{aligned} \mathbf{F}_1 &= \begin{pmatrix} 1 + \frac{u}{l} & -\frac{u}{h} \\ \frac{v}{l} & 1 - \frac{v}{h} \end{pmatrix}, & \mathbf{F}_2 &= \begin{pmatrix} 1 + \frac{u}{l} & \frac{u}{h} \\ \frac{v}{l} & 1 + \frac{v}{h} \end{pmatrix}, \\ \mathbf{F}_3 &= \begin{pmatrix} 1 - \frac{u}{L} & \frac{u}{h} \\ -\frac{v}{L} & 1 + \frac{v}{h} \end{pmatrix}, & \mathbf{F}_4 &= \begin{pmatrix} 1 - \frac{u}{L} & -\frac{u}{h} \\ -\frac{v}{L} & 1 - \frac{v}{h} \end{pmatrix}, \end{aligned} \quad (2.86)$$

where l and L are the horizontal lengths of elements to the left and to the right of the interface, respectively, and h is the common height of all elements (see Figure 2.17). By using element areas $A_1 = A_2 = h l/2$ and $A_3 = A_4 = h L/2$, (2.86) also implies that

$$\begin{aligned} \sum_{e=1}^4 \mathbf{F}_e A_e &= \mathbf{I} \sum_{e=1}^4 A_e \quad \Rightarrow \quad \sum_{e=1}^4 \frac{\partial \mathbf{F}_e}{\partial u} A_e = 0, \\ \wedge \quad \sum_{e=1}^4 \frac{\partial \mathbf{F}_e}{\partial v} A_e &= 0 \quad \Leftrightarrow \quad \sum_{e=1}^4 \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0} A_e = \mathbf{0}. \end{aligned} \quad (2.87)$$

We note that these relations also hold if we do not start from the undeformed ground state but if instead we start from an affinely-deformed initial state with uniform deformation gradient $\mathbf{F}_0 = \text{const.}$ In this case deformation gradients in (2.86) change into $\hat{\mathbf{F}}_i = \mathbf{F}_0 \mathbf{F}_i$ with \mathbf{F}_i from (2.86), so that one arrives at the same final relation in (2.87).

Let us assume the energy E_i of any atom i is defined by an interatomic potential of the Embedded Atom Method [Daw and Baskes, 1984], so that (2.17) applies. The inner-element sampling atoms experience an affine deformation. For concision, let us denote by \mathcal{S} the set of all lattice sites contained within a neighborhood of interaction, and \mathbf{R}_j be the distance vector from any atom to its neighboring lattice site j ($j \in \mathcal{S}$). For an EAM potential this leads to

$$E_{e,\text{inner}} = \sum_{j \in \mathcal{S}} \frac{1}{2} \Phi(|\mathbf{F}_e \mathbf{R}_j|) + \mathcal{F}(\rho_e), \quad \rho_e = \sum_{j \in \mathcal{S}} \phi(|\mathbf{F}_e \mathbf{R}_j|). \quad (2.88)$$

Consequently, we obtain the contribution to the residual force from the sampling atom within element e via

$$\frac{\partial E_{e,\text{inner}}}{\partial \mathbf{x}_0} = \frac{\partial E_{e,\text{inner}}}{\partial \mathbf{F}_e} \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0} = \sum_{j \in \mathcal{S}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_e \mathbf{R}_j|) + \mathcal{F}'(\rho_e) \phi'(|\mathbf{F}_e \mathbf{R}_j|) \right] \frac{\mathbf{F}_e \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_e \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0}. \quad (2.89)$$

Similarly, we can consider sampling atoms on element edges which are assumed to experience an affine deformation on either side of the interface. Thus, the energy of a sampling atom on the edge between elements a and b involves half an atomic neighborhood $\mathcal{S}_{1/2}$ within either of the two adjacent elements. In addition, for centrosymmetric lattices, summation over any (arbitrarily oriented) semicircle of interaction $\mathcal{S}_{1/2}$ yields the same answer, viz.

$$\begin{aligned} E_{a-b,\text{edge}} &= \sum_{j \in \mathcal{S}_{1/2}} \frac{1}{2} \Phi(|\mathbf{F}_a \mathbf{R}_j|) + \sum_{j \in \mathcal{S}_{1/2}} \frac{1}{2} \Phi(|\mathbf{F}_b \mathbf{R}_j|) + \mathcal{F}(\rho_e), \\ \rho_e &= \sum_{j \in \mathcal{S}_{1/2}} \phi(|\mathbf{F}_a \mathbf{R}_j|) + \sum_{j \in \mathcal{S}_{1/2}} \phi(|\mathbf{F}_b \mathbf{R}_j|). \end{aligned} \quad (2.90)$$

Here, the contribution to the residual force depends on the location of the sampling atom. In general, we have

$$\begin{aligned} \frac{\partial E_{a-b,\text{edge}}}{\partial \mathbf{x}_0} &= \frac{\partial E_{a-b,\text{edge}}}{\partial \mathbf{F}_a} \cdot \frac{\partial \mathbf{F}_a}{\partial \mathbf{x}_0} + \frac{\partial E_{a-b,\text{edge}}}{\partial \mathbf{F}_b} \cdot \frac{\partial \mathbf{F}_b}{\partial \mathbf{x}_0} \\ &= \sum_{j \in \mathcal{S}_{1/2}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_a \mathbf{R}_j|) + \mathcal{F}'(\rho_e) \phi'(|\mathbf{F}_a \mathbf{R}_j|) \right] \frac{\mathbf{F}_a \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_a \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_a}{\partial \mathbf{x}_0} \\ &\quad + \sum_{j \in \mathcal{S}_{1/2}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_b \mathbf{R}_j|) + \mathcal{F}'(\rho_e) \phi'(|\mathbf{F}_b \mathbf{R}_j|) \right] \frac{\mathbf{F}_b \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_b \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_b}{\partial \mathbf{x}_0}. \end{aligned} \quad (2.91)$$

If both elements a and b lie within the patch of four elements, then the above form holds as written.

If element a or b lies outside the patch of elements, then the respective term in (2.91) vanishes since $\partial \mathbf{F}_e / \partial \mathbf{x}_0 = \mathbf{0}$ if element e is not adjacent to node 0. When considering residual forces in an affinely-strained ground state with uniform deformation gradient \mathbf{F}_0 , then relations (2.59) and (2.91) hold if we set $\mathbf{F}_e = \mathbf{F}_0$ for all $e = 1, \dots, 4$. For example, the case of an affine ground state reduces (2.91) to

$$\begin{aligned} \left. \frac{\partial E_{a-b, \text{edge}}}{\partial \mathbf{x}_0} \right|_{\mathbf{F}_e = \mathbf{F}_0} &= \sum_{j \in \mathcal{S}_{1/2}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|) \right] \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \cdot \left(\frac{\partial \mathbf{F}_a}{\partial \mathbf{x}_0} + \frac{\partial \mathbf{F}_b}{\partial \mathbf{x}_0} \right), \\ \rho_0 &= \sum_{j \in \mathcal{S}} \phi(|\mathbf{F}_0 \mathbf{R}_j|). \end{aligned} \quad (2.92)$$

In contrast, if the sampling atom is located on an edge on the perimeter of the patch of elements, then only one of the two semicircles of interaction is affected by \mathbf{x}_0 (assume it is element a), which yields for the affine ground state

$$\begin{aligned} \left. \frac{\partial E_{a-b, \text{edge}}}{\partial \mathbf{x}_0} \right|_{\mathbf{F}_e = \mathbf{F}_0} &= \sum_{j \in \mathcal{S}_{1/2}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|) \right] \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_a}{\partial \mathbf{x}_0}, \\ \rho_0 &= \sum_{j \in \mathcal{S}} \phi(|\mathbf{F}_0 \mathbf{R}_j|). \end{aligned} \quad (2.93)$$

Altogether, for the shown patch of four elements the residual force on the central repatom 0 in an affinely deformed configuration with uniform deformation gradient \mathbf{F}_0 is given by summing the contribution from all $E_{i, \text{inner}}$ and $E_{i, \text{edge}}$ (as derived above) as well as those contributions from $E_{i, \text{vertex}}$, which results in

define $A_j = \frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|)$

$$\begin{aligned}
\tilde{\mathbf{F}}_0(\mathbf{x}_0) = & - \sum_{e=1}^4 w_{e,\text{inner}} \sum_{j \in \mathcal{S}} A_j \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{F}_0 \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0} \\
& - \sum_{e,f} w_{e-f,\text{edge (inner)}} \sum_{j \in \mathcal{S}_{1/2}} A_j \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{F}_0 \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \cdot \left(\frac{\partial \mathbf{F}_{e,1}}{\partial \mathbf{x}_0} + \frac{\partial \mathbf{F}_{f,2}}{\partial \mathbf{x}_0} \right) \\
& - \sum_{e=1}^4 w_{e,\text{edge (perimeter)}} \sum_{j \in \mathcal{S}_{1/2}} A_j \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{F}_0 \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0} \\
& - \left(w_4 \sum_{\mathbf{R}_j \in [0^\circ; \alpha)} + w_1 \sum_{\mathbf{R}_j \in [\alpha; \pi/2)} + w_0 \sum_{\mathbf{R}_j \in [\pi/2; \pi)} \right) A_j \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{F}_0 \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_1}{\partial \mathbf{x}_0} \\
& - \left(w_0 \sum_{\mathbf{R}_j \in [0^\circ; \pi/2)} + w_1 \sum_{\mathbf{R}_j \in [\pi/2; \pi-\beta)} + w_2 \sum_{\mathbf{R}_j \in [\pi-\beta; \pi)} \right) A_j \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{F}_0 \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_2}{\partial \mathbf{x}_0} \\
& - \left(w_2 \sum_{\mathbf{R}_j \in [0^\circ; \beta)} + w_3 \sum_{\mathbf{R}_j \in [\beta; \pi/2)} + w_0 \sum_{\mathbf{R}_j \in [\pi/2; \pi)} \right) A_j \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{F}_0 \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_3}{\partial \mathbf{x}_0} \\
& - \left(w_0 \sum_{\mathbf{R}_j \in [0^\circ; \pi/2)} + w_4 \sum_{\mathbf{R}_j \in [\pi/2-\alpha; \pi/2)} + w_3 \sum_{\mathbf{R}_j \in [\pi/2; \pi/2+\alpha)} \right) A_j \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{F}_0 \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_4}{\partial \mathbf{x}_0},
\end{aligned} \tag{2.94}$$

where $\mathbf{R}_j \in [x; y)$ denotes the set of all lattice sites contained within the segment of the circle of interaction ranging from angle x to angle y (counter-clockwise from the horizontal axis) and

$$\alpha = \arctan(h/l), \quad \beta = \arctan(h/L). \tag{2.95}$$

In the new optimal first- and second-order summation rules, we choose weights according to

$$w_0 = w_1 = \dots = w_{n_e} = w_{\text{rep}}, \tag{2.96}$$

i.e., we assign equal weights $w_i = w_{\text{rep}}$ to all sampling atoms on repatoms ($i = 0, \dots, 4$) so that (2.94) reduces to

$$\begin{aligned}
\tilde{\mathbf{F}}_0(\mathbf{x}_0) = & - \sum_{e=1}^4 \left[w_{e,\text{inner}} + \frac{1}{2} w_{e,\text{edge (perimeter)}} + \sum_{i=1}^2 \frac{1}{2} w_{e,i,\text{edge (inner)}} + \frac{1}{2} w_{\text{rep}} \right] \\
& \times \sum_{j \in \mathcal{S}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|) \right] \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \cdot \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0},
\end{aligned} \tag{2.97}$$

where $w_{e,i,\text{edge (inner)}}$ denotes the weight of the sampling atom located on the (inner) edge of element e shared with adjacent element e .

As discussed earlier, the undeformed ground state ($\mathbf{F}_0 = \mathbf{I}$) of an infinite single crystal is characterized by a minimum of the total potential energy. This implies that

$$\sum_{j \in \mathcal{S}} \left[\frac{1}{2} \Phi'(|\mathbf{R}_j|) + \mathcal{F}'(\rho_0(\mathbf{I})) \phi'(|\mathbf{R}_j|) \right] \frac{\mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{R}_j|} = \mathbf{0}. \quad (2.98)$$

We see that application of $\mathbf{F}_0 = \mathbf{I}$ to (2.97) makes the residual force on the interface repatom in the undeformed ground state (using the new summation rules) vanish for all choices of l and L . Any other summation rule (including nodal, cluster, or quadrature rules) whose weights at repatom locations are based on element sizes or geometries does not allow for this reduction, so that they will necessarily produce residual force artifacts. For example, consider a nodal summation rule with $w_{e,\text{inner}} = w_{i,\text{edge}} = 0$ applied to the chosen geometry in an arbitrary affinely-strained configuration (with deformation gradient \mathbf{F}_0). In this case, insertion of the appropriate nodal weights leads to

$$\tilde{\mathbf{F}}_0(\mathbf{x}_0) = -2w_r \frac{L-l}{lL} f(\mathbf{R}_1, \dots, \mathbf{R}_n; \mathbf{F}_0) \quad (2.99)$$

with a generally non-vanishing function $f(\cdot)$. Apparently, this force only vanishes if $l = L$, i.e., in a uniform QC mesh. Otherwise, a residual/spurious force artifact (in the horizontal direction) appears at the interface.

Finally, the new first- and second-order summation rules are designed to eliminate not only residual forces in the undeformed ground state but also all spurious force artifacts in any affinely-deformed configuration in centrosymmetric lattices. To this end, weights are chosen such that

$$w_{e,\text{inner}} + \frac{1}{2}w_{e,\text{edge(perimeter)}} + \sum_{i=1}^2 \frac{1}{2}w_{e,i,\text{edge(inner)}} + \frac{1}{2}w_{\text{rep}} = 0 \quad (2.100)$$

for all elements e . To minimize the energy error, this is achieved by defining $w_{\text{rep}} = \pi r^2$, where r denotes the effective atomic interaction distance. In the first-order rule $(0, 1^*)$ we have $w_{e,\text{edge}} = 0$, and in the second-order rule $(0, 1^*, 4^*)$ we define $w_{e,\text{edge}} = r(L_e - 2r)$ with L_e being the length of element edge e ; see also the graphical interpretation in Figure 3.30. In either of the two rules, the inner-element sampling atom's weight is set to

$$w_{e,\text{inner}} = n_{e,\text{total}} - \left(\frac{1}{2}w_{\text{rep}} + \frac{1}{2}w_{e,\text{edge(perimeter)}} + \sum_{i=1}^2 \frac{1}{2}w_{e,i,\text{edge(inner)}} \right). \quad (2.101)$$

That is, after assigning weights to all repatoms and to all sampling atoms on edges, the inner-element sampling atoms receive the remaining number of lattice sites within the respective elements as their

weights (and $n_{e,\text{total}}$ is proportional to the element area A_e). This leads to a spurious force of

$$\tilde{\mathbf{F}}_0(\mathbf{x}_0) = - \sum_{j \in \mathcal{S}} \left[\frac{1}{2} \Phi'(|\mathbf{F}_0 \mathbf{R}_j|) + \mathcal{F}'(\rho_0) \phi'(|\mathbf{F}_0 \mathbf{R}_j|) \right] \frac{\mathbf{F}_0 \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F}_0 \mathbf{R}_j|} \cdot \sum_{e=1}^4 A_e \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_0} = \mathbf{0}. \quad (2.102)$$

Hence, in the undeformed ground state with $\mathbf{F}_0 = \mathbf{I}$ the first sum vanishes, whereas in any affinely-deformed state with $\mathbf{F}_0 \neq \mathbf{I}$ the second sum vanishes.

In summary, the new first- and second-order summation rules **do not produce any residual or spurious** force artifacts on the interface repatoms in **any affinely-deformed, non-uniform** QC mesh.

2.3.2.5 Cauchy-Born kinematics in the nonlocal framework

Depending on the criterion chosen for when a Cauchy-Born sampling atom is used in the new summation rules, it is possible that the neighbors of the sampling atom might extend beyond the limits of the containing element. Shape functions are commonly used to interpolate lattice site positions within elements, but within the described nonlocal framework there must also be a way to compute local neighborhoods for Cauchy-Born sampling atoms in small elements whose actual atomic neighborhoods of inner-element sampling atoms would reach into adjacent elements. The following analysis shows that shape functions can also be used to extrapolate lattice site positions outside the element to construct a local neighborhood deforming according to the Cauchy-Born rule, even for small elements.

Consider an atomic neighborhood containing n lattice sites centered around a sampling atom located at the origin. According to the Cauchy-Born rule, we assume affine kinematics and therefore write the potential energy of the sampling atom as

$$V = \sum_{j=1}^n \frac{1}{2} \Phi(|\mathbf{F} \mathbf{R}_j|) + \mathcal{F}(\rho), \quad \rho = \sum_{j=1}^n \phi(|\mathbf{F} \mathbf{R}_j|) \quad (2.103)$$

with deformation gradient

$$\mathbf{F}(\mathbf{X}) = \sum_{\alpha=1}^4 \mathbf{x}^\alpha \otimes \text{Grad } N^\alpha(\mathbf{X}) \quad (2.104)$$

to be evaluated at the (fictitious or real) site of the sampling atom. \mathbf{x}^α are the deformed positions of the simplicial element's nodes (repatoms) and $N^\alpha(\mathbf{X})$ denote the corresponding shape functions. Consequently, we know that

$$\frac{\partial V}{\partial \mathbf{x}_a} = \frac{\partial V}{\partial \mathbf{F}} \cdot \frac{\partial \mathbf{F}}{\partial \mathbf{x}_a}, \quad (2.105)$$

where

$$\frac{\partial V}{\partial \mathbf{F}} = \mathbf{P} = \sum_{j=1}^n \left[\frac{1}{2} \Phi'(|\mathbf{F} \mathbf{R}_j|) + \mathcal{F}'(\rho) \phi'(|\mathbf{F} \mathbf{R}_j|) \right] \frac{\mathbf{F} \mathbf{R}_j \otimes \mathbf{R}_j}{|\mathbf{F} \mathbf{R}_j|} \quad (2.106)$$

is the first Piola-Kirchhoff stress tensor at the location of the sampling atom [Tadmor et al., 1996].

In addition,

$$\frac{\partial F_{iJ}}{\partial x_n^k} = \delta_{in} (\text{Grad } N_k)_J, \quad (2.107)$$

so that with $\mathbf{r}_j = \mathbf{F}\mathbf{R}_j$ the force contribution is given by

$$\mathbf{F}_k = -\frac{\partial V}{\partial \mathbf{x}_k} = -\sum_{j=1}^n \left[\frac{1}{2} \Phi'(|\mathbf{r}_j|) + \mathcal{F}'(\rho) \phi'(|\mathbf{r}_j|) \right] \frac{\mathbf{r}_j}{|\mathbf{r}_j|} (\mathbf{R}_j \cdot \text{Grad } N_k). \quad (2.108)$$

Since in simplicial elements shape functions vary linearly in space, their gradients are constant.

Given the shape function N_k value at two distinct locations \mathbf{X}_i and \mathbf{X}_j , we thus know that

$$\text{Grad } N_k = \frac{N_k(\mathbf{X}_j) - N_k(\mathbf{X}_i)}{(\mathbf{X}_j - \mathbf{X}_i)_n} \mathbf{e}_n, \quad (2.109)$$

where $(\mathbf{X}_j - \mathbf{X}_i)_n$ denotes the n -th component of vector $\mathbf{X}_j - \mathbf{X}_i$ and \mathbf{e}_n is the n -th base vector.

Therefore,

$$\mathbf{R}_j \cdot \text{Grad } N_k = N_k(\mathbf{X}_j) - N_k(\mathbf{X}_i) \quad (2.110)$$

and overall we arrive at

$$\mathbf{F}_k = -\frac{\partial V}{\partial \mathbf{x}_k} = -\sum_{j=1}^n \left[\frac{1}{2} \Phi'(|\mathbf{r}_j|) + \mathcal{F}'(\rho) \phi'(|\mathbf{r}_j|) \right] \frac{\mathbf{r}_j}{|\mathbf{r}_j|} [N_k(\mathbf{X}_j) - N_k(\mathbf{X}_i)], \quad (2.111)$$

which is exactly the same structure used in the nonlocal QC formulation, cf. (2.19). In other words, application of the Cauchy-Born rule requires to construct a local neighborhood of the sampling atom which is assumed to entirely lie within the same element as the sampling atom. This results in interpolation within the element and extrapolation outside the element.

2.3.2.6 Derivation of the average deformation for a discrete patch of simplicial elements

The development of the new summation rule required relation (2.78). This relation can be shown for an arbitrary patch of simplicial (triangular) elements as follows:

Each element has shape functions $N_i^e(\mathbf{X})$ with $i = 1, 2, 3$ associated with its three nodes (numbered in counter-clockwise order). Thus, we obtain

$$\mathbf{F}_e = \sum_{i=1}^3 \mathbf{x}_i \otimes \text{Grad } N_i^e \quad (2.112)$$

so that (using indicial notation)

$$\frac{\partial}{\partial(\mathbf{x}_1)_k}(\mathbf{F}_e)_{iJ} = \delta_{ik} (\text{Grad } N_1^e)_J. \quad (2.113)$$

For the chosen linear triangular element we can choose the node numbering such that $N_1^e = r$, $N_2^e = s$, and $N_3^e = 1 - r - s$ with barycentric coordinates r and s . Consequently, $\partial N_1^e / \partial r = 1$ and $\partial N_1^e / \partial s = 0$ in each element. Furthermore,

$$\text{Grad } N_1^e = \begin{pmatrix} \frac{\partial r}{\partial x} & \frac{\partial s}{\partial x} \\ \frac{\partial r}{\partial y} & \frac{\partial s}{\partial y} \end{pmatrix} \begin{pmatrix} \frac{\partial N_1^e}{\partial r} \\ \frac{\partial N_1^e}{\partial s} \end{pmatrix}. \quad (2.114)$$

By the inverse function theorem,

$$\begin{pmatrix} \frac{\partial r}{\partial x} & \frac{\partial s}{\partial x} \\ \frac{\partial r}{\partial y} & \frac{\partial s}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{pmatrix}^{-1} = \begin{pmatrix} x_1 - x_3 & y_1 - y_3 \\ x_2 - x_3 & y_2 - y_3 \end{pmatrix}^{-1} \quad (2.115)$$

so that

$$\begin{aligned} \text{Grad } N_1 &= \begin{pmatrix} \frac{\partial r}{\partial x} & \frac{\partial s}{\partial x} \\ \frac{\partial r}{\partial y} & \frac{\partial s}{\partial y} \end{pmatrix} \begin{pmatrix} \frac{\partial N_1^e}{\partial r} \\ \frac{\partial N_1^e}{\partial s} \end{pmatrix} \\ &= \frac{1}{A_e} \begin{pmatrix} y_2 - y_3 & -(y_1 - y_3) \\ -(x_2 - x_3) & x_1 - x_3 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{A_e} \begin{pmatrix} y_2 - y_3 \\ x_3 - x_2 \end{pmatrix}. \end{aligned} \quad (2.116)$$

Now, consider a patch of elements with a common center node which is labeled node 1 in each element e for convenience, so that nodes 2 and 3 of each element lie on the perimeter of the patch of elements. Then, it follows that

$$\sum_{e=1}^{n_e} A_e \text{Grad } N_1^e = \sum_{e=1}^{n_e} \begin{pmatrix} y_2^e - y_3^e \\ x_3^e - x_2^e \end{pmatrix} = \mathbf{0}. \quad (2.117)$$

The sum vanishes because the sum of all vectors along the perimeter of the patch must form a closed loop with a net vector of zero magnitude. Finally, along with (2.113) this confirms (2.78) for a discrete patch of elements, viz. that

$$\frac{\partial}{\partial \mathbf{x}_1} \sum_{e=1}^{n_e} A_e \mathbf{F}_e = \sum_{e=1}^{n_e} \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_1} A_e = \mathbf{0}. \quad (2.118)$$

2.3.3 Summary and conclusions

With regards to summation rules our analysis has shown the following:

- In one dimension, nodal, cluster, and quadrature summation rules do not produce any residual forces in the undeformed ground state if elements are sufficiently large so that spheres of interaction around repatoms do not overlap. The same, of course, holds true in the full atomistic limit (in any dimensions).
- In one and two dimensions, nodal, cluster, and quadrature summation rules produce spurious forces in an affinely-deformed non-uniform QC mesh. Of the popular summation rules only element quadrature (or local QC) avoids spurious force artifacts in sufficiently large elements. When using quadrature summation rules, residual forces vanish in the large-element limit, if all sampling atoms at repatom locations are assigned the same weights.
- New summation rules have been introduced which assign the same weight to all sampling atoms at repatom locations. In the large-element limit this choice guarantees no force artifacts in both the undeformed configuration and in any affinely-deformed state in one and two dimensions. In three dimensions, force artifacts can appear which are small compared to previous summation rules and remain negligible as long as the element aspect ratios remain small (which should be the case in any reasonable QC mesh).
- The new summation rules make use of the Cauchy-Born rule at the quadrature-type sampling atom (in analogy to the traditional local/nonlocal QC method). This removes residual forces at mesh interfaces between larger elements and reduces force artifacts to an even smaller region of elements right above the atomistic limit, which will be confirmed in chapter 3.

2.4 Extension to three dimensions and the full range of element sizes

In three dimensions, the analysis is analogous to the two-dimensional one presented above with one crucial difference. In two dimensions, we could devise an optimal summation rule with vanishing force artifacts and minimal approximation errors because – with the proper choice of summation rule weights – the contributions from all inner angles of each triangular element recombined into a semi-circle of interaction, cf. (2.70). In three dimensions, the four solid angles within each tetrahedral element unfortunately do *not* recombine into a semi-sphere of interaction in general. In fact, the sum of all four solid angles in a tetrahedron is not constant and depends on the exact geometry; see for example [Masket, 1957]. For example, let α_i ($i = 1, \dots, 6$) denote the six dihedral angles between pairs of tetrahedral faces. The sum of all four solid angles ϕ_i ($i = 1, \dots, 4$) amounts to

$$\sum_{i=1}^4 \phi_i = 2 \sum_{i=1}^6 \alpha_i (\text{rad}) - 4 \pi (\text{sr}). \quad (2.119)$$

Consider a regular tetrahedron in which all dihedral angles are $\arccos(1/3)$, so that the above sum of all four solid angles yields $12\arccos(1/3) - 4\pi \approx 2.21$. Now, let the height of the tetrahedron shrink to zero, in which case the sum of all four solid angles must approach that of a hemisphere, viz. 2π . Ergo, the sum of all four solid angles within a tetrahedron is geometry-dependent, which is why the neighboring lattice sites of the four repatoms within an element do not recombine into a hemisphere of interaction. Spurious and residual forces will thus depend on the specific element geometry and there exists no general quadrature summation rule that removes all residual forces. However, any reasonable QC discretization will aim to (and any useful remeshing technique should) avoid large element condition numbers. In this case, we can accept small errors and extend the new summation rule to 3D.

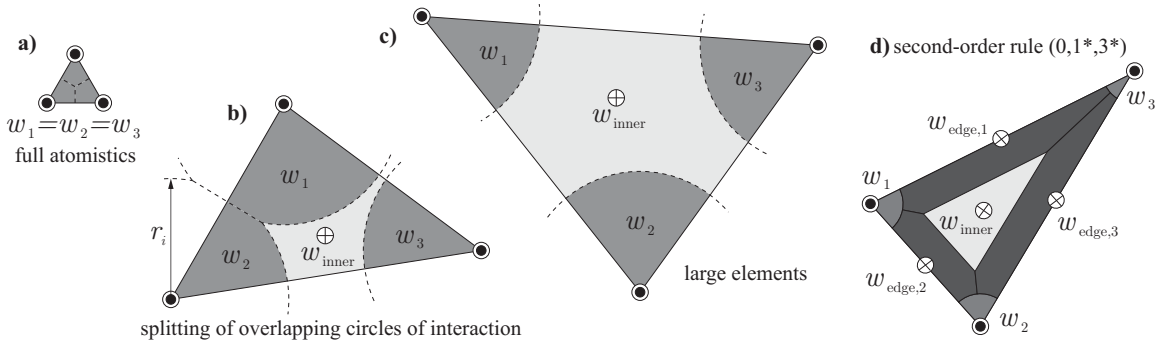


Figure 2.18: Schematic illustration of the weight computation for arbitrary element sizes using summation rule $(0,1^*)$: a) fully atomistic (all sampling atoms have the same weight $w_i = 1$); b) overlapping circles of interaction in intermediate-sized elements are split to determine weights; c) in large elements sampling atoms at element vertices have the same weights again. The inner-element quadrature atom can either be treated by the Cauchy-Born rule or by applying its actual atomic neighborhood (which may reach into adjacent elements if elements are not large). d) Second-order summation rule $(0,1^*,3^*)$.

New summation rules in three dimensions As before, every sampling atom at the location of a repatom is assigned the same weight, now $w_{\text{rep}} = 4\pi/3 r_{\text{eff}}^3 \rho$, which represents the number of lattice sites within the effective interaction distance of the atomic potential. In addition, each element receives one quadrature-type sampling atom feeling affine neighborhood changes in large elements. When using fictitious sampling atoms and the Cauchy-Born rule, this scheme agrees with the classical local QC model [Tadmor et al., 1996] enriched by sampling atoms at element vertices. This analogy, however, breaks down with decreasing element size where, ultimately, the original QC formulation would switch to nonlocal QC while we maintain the same nonlocal structure.

To this end, we generalize the summation rule $(0,1^*)$ to elements of arbitrary sizes. Similar in spirit to the truncation of overlapping clusters [Knap and Ortiz, 2001, Eidel and Stukowski, 2009], we define sampling atom weights as follows. When effective interaction spheres of sampling atoms on element vertices overlap, their weights are obtained by removing the overlapping regions and

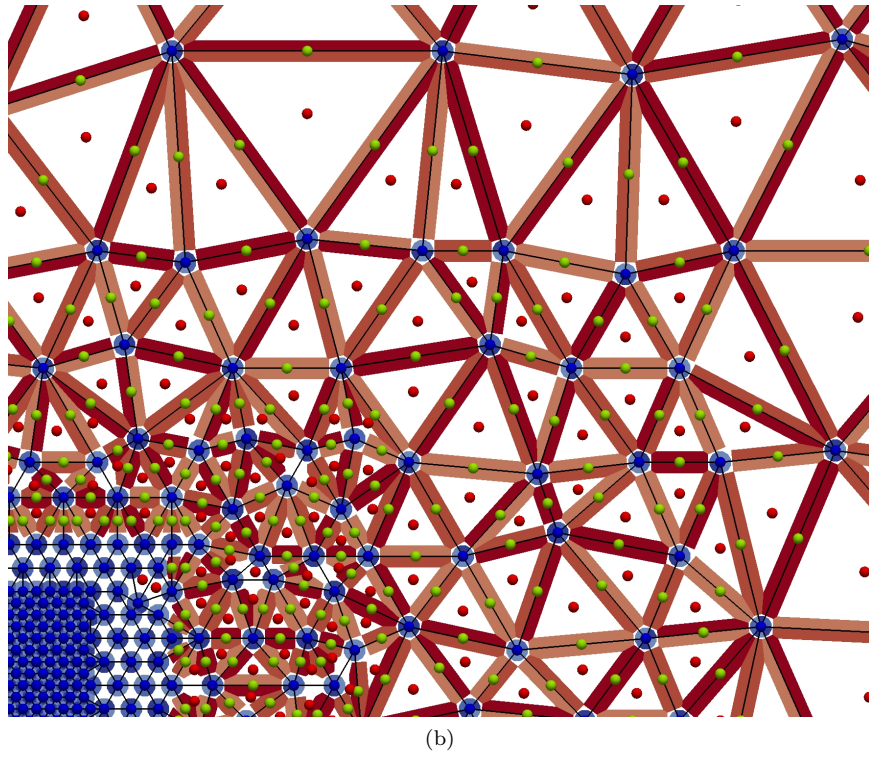
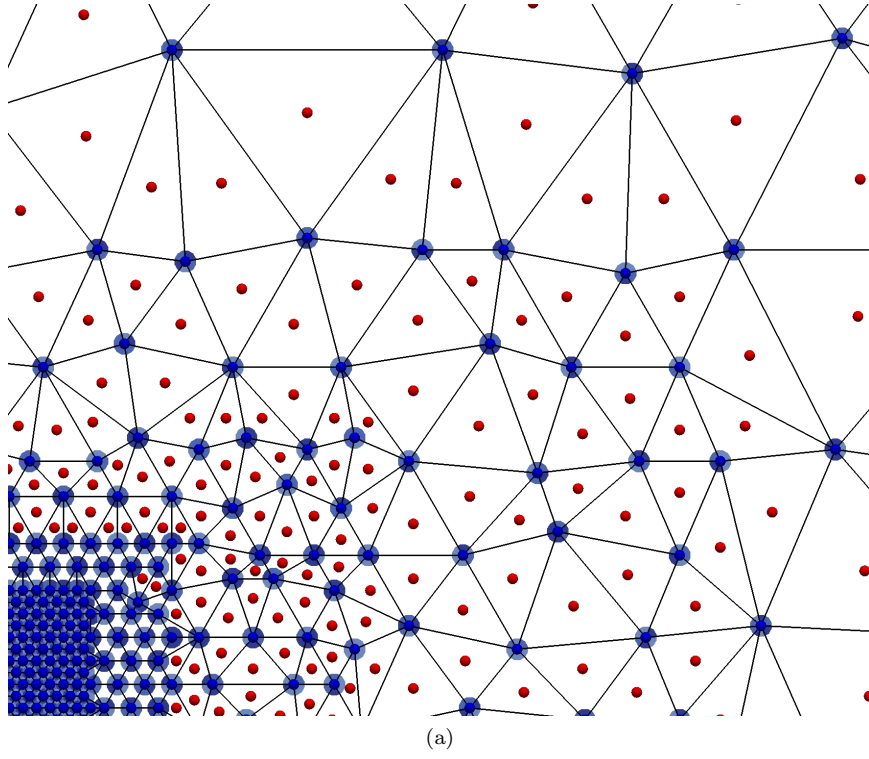


Figure 2.19: Illustration of sampling atom weights for a) the first-order rule $(0, 1^*)$ and b) the second-order rule $(0, 1^*, 3^*)$.

splitting weights equally, cf. Figure 2.18. As before, the inner-element sampling atom's weight is given by the remaining lattice sites according to

$$w_{e,\text{inner}} = V_e - \sum_{i=1}^4 w_i \geq 0 \quad (2.120)$$

with V_e the total number of lattice sites in element e . This three-dimensional $(0, 1^*)$ scheme locally reduces to full atomistics as the element size decreases. Whether or not the inner-element sampling atom should be computed using its actual atomic neighborhood or a fictitious Cauchy-Born neighborhood will be discussed by the aid of numerical examples in Section 3.2.2. We will see that the Cauchy-Born rule is generally better suited to represent the collections of inner-element lattice sites rather than picking a specific lattice site and its actual neighborhood. This can be expected from our analyses in one and two dimensions which demonstrated vanishing residual and spurious forces in affinely deformed crystals if the inner-element sampling atoms experienced affine neighborhood deformations (which would not be the case when picking actual lattice sites so that neighbors reach into adjacent elements). Figure 2.19 shows the sampling atom weights of a non-uniform QC mesh in two dimensions for a non-uniform 2D mesh that is gradually coarse-grained around an atomistic region in the bottom left corner. Blue circles denote weights of sampling atoms at repatom locations, white regions represent weights assigned to inner-element sampling atoms, and red regions show weights of sampling atoms on edges in the second-order rule. Small dots illustrate sampling atom locations (blue are those on repatom sites, red at barycenters, and green are those at edge midpoints). The 3D scenario is analogous but hard to visualize; visualizations of the implemented technique are shown in Section 4.5.3.

We note that this summation rule automatically satisfies zeroth-order consistency, cf. (2.23a), because all sampling atom weights add up to the total number of lattice sites. As long as elements are sufficiently large (or fully atomistic), the scheme also satisfies first-order consistency (2.23b). The same applies in the limit of full atomistics. However, there is a narrow intermediate regime of element sizes on the order of the interatomic interaction radius where first-order consistency is not guaranteed.

Similarly to 2D, we can devise higher-order summation rules, such as the second-order rule $(0, 1^*, 4^*)$ with one additional sampling atom on each element face, or the third-order scheme $(0, 1^*, 4^*, 6^*)$ with one additional sampling atom on each element face and edge. However, the latter becomes computationally quite expensive and our numerical investigations in this thesis will demonstrate the benefits of the first- and second-order summation rules.

2.5 Summary and conclusions

In short, we have developed a novel, fully-nonlocal, energy-based QC formulation which requires no distinction between, or special treatment of, atomistic, transition, and coarse regions. For comparison and experimentation, we have captured existing summation rules in a unified sampling atom framework based on a small set of sampling atoms which include the repatoms (in order to seamlessly bridge to full atomistics) and additional representative lattice sites. We have analyzed, investigated, and compared various existing summation rules in one and two dimensions. We have derived and analyzed in one and two dimensions a family of new, “optimal” summation rules which, for sufficiently-large elements, do not exhibit spurious force artifacts in non-uniform QC meshes. We have extended the family of summation rules for three dimensions and for arbitrarily-sized elements, though each of those extensions loses some optimality. In the next chapter we will use this framework first to numerically compare the summation rules for particular, isolated configurations of a body, after which we will perform full simulations by equilibrating boundary value problems with external forces and boundary conditions.

In all subsequent examples of this thesis, we use the above framework to simulate the response of crystalline ensembles at zero temperature. Finite temperature can be accounted for by the QC method and can be added on top of the spatial coarse-graining techniques discussed here; see for example [Shenoy et al., 1999a, Dupuy et al., 2005, Kulkarni et al., 2008, Marian et al., 2010, Ariza et al., 2012, Tadmor et al., 2013]. In addition, we only consider quasistatic (low-strain-rate) phenomena so that inertial effects are negligible. Equilibration is performed using the inertial relaxation algorithm FIRE of Bitzek et al. [2006].

Chapter 3

Verification and Validation

3.1 Overview

In this chapter we investigate summation rules for the fully-nonlocal energy-based QC method with respect to their accuracy and efficiency in coarse-grained atomistic simulations. Most previous studies have presented instructive albeit simple one- or two-dimensional examples, oftentimes with simplistic interatomic potentials. Here, we report the first comparative survey of summation rules applied to large-scale two- and three-dimensional simulations with uniform and completely-random meshes involving thousands to tens of millions of degrees of freedom with realistic interatomic potentials.

We employ a general formulation of summation rules based on sampling atoms which differ by their locations and weights to approximate the total Hamiltonian of the crystalline atomistic ensemble. In particular, we include traditional cluster and quadrature-type summation rules as well as the previously-introduced first- and second-order sampling rules which exhibit minimal spurious force artifacts in affinely-deformed meshes.

We start by investigating the performance of the various summation rules with calculations of energies and residual forces in specific deformed configurations, i.e., without equilibration. Our benchmark simulations with equilibration include the calculation of elastic constants from completely random QC meshes and the inhomogeneous deformation of aggressively coarse-grained crystals containing nano-voids. In the elastic regime, we directly compare QC results to those of full atomistics in order to not only quantify approximation errors but also to assess local errors in complex QC simulations. Going beyond elasticity, we illustrate the performance of the fully-nonlocal energy-based QC method with the new second-order summation rule by nanoindentation examples in two and three dimensions with automatic mesh adaptation, thereby illustrating the formation and evolution of lattice defects in the QC method and investigating the effective response. Lastly, we investigate the failure of a hole in a three-dimensional plate using the new second-order summation rule by observing the microstructure formation around the hole using implicit dynamics.

Because of the difficulty in representing the energy of the atoms on the faces of surface elements, scenarios with free surfaces are handled poorly by all previously-existing summation rules. Using the new formulation, we investigate several problems which involve free surfaces, including surface energies and elastic moduli, through which it will be seen that the new second-order summation rule qualifies to accurately and efficiently coarse-grain free surfaces and interfaces when these coincide with element faces in three dimensions (or edges in two dimensions).

Though the formulation and implementation are quite capable of 3D calculations and many 3D examples will be shown, we deliberately include some two-dimensional examples for better visibility of the phenomena of interest. Overall, our findings provide guidelines for the selection of summation rules for the fully-nonlocal energy-based QC method and confirm the superior performance of the new summation rules.

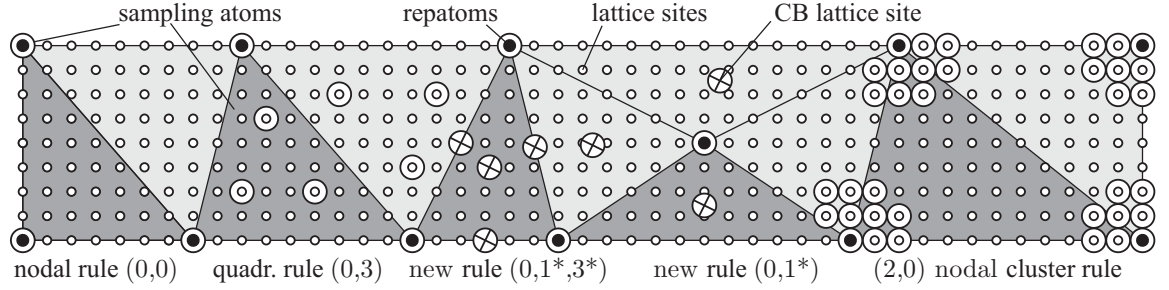


Figure 3.1: Schematic illustration of the QC discretization and different types of summation rules, including quadrature and cluster summation rules as well as the new summation schemes of first and second order (CB denotes the Cauchy-Born rule applied to quadrature-type sampling atom).

In all examples, we compare the following types of summation rules (Figure 3.1 schematically shows the different rules):

- $(0,0)$: a purely nodal summation rule,
- $(n_{Cl},0)$: nodal cluster summation rules with n_{Cl} shells of clusters,
- $(0,n_Q)$: quadrature summation rules with n_Q quadrature-type sampling atoms per element,
- $(0,1^*)$: the new first-order summation rule,
- $(0,1^*,4^*)$: the new second-order summation rule.

Note that we exclude pure element quadrature rules from our analysis, since they require a switching mechanism from the local element-based formulation to the nonlocal node-based description in the limit of full atomistics. Here, we focus only on fully-nonlocal QC schemes that seamlessly bridge across all element sizes. Furthermore, in the case of quadrature summation rules, general three-dimensional examples require numerical compromises: quadrature summation rules define their sampling atom locations within elements as lattice sites closest to Gauss-Lobatto quadrature points

(commonly used in the finite element method). Of course, the existence of a lattice constrains the choice of quadrature-atom locations. In the following, if several lattice sites are located at the same distance from a quadrature point, we make a random choice. If no such lattice site exists within an element (or if the only ones coincide with already existing sampling atoms), then no new sampling atom is inserted. Also, for higher-order quadrature rules we identify quadrature-type sampling atoms within each element in a random order.

3.2 Examinations of isolated configurations

Let us demonstrate the performance of the new summation rule for the fully-nonlocal energy-based QC formulation by a few selected examples in two and three dimensions. In this section, we focus on the more difficult case of non-uniform meshes (as opposed to uniform) and we will examine energy and force errors in an isolated configuration, i.e., no solver is being used and no equilibration is being performed. Simulations with boundary conditions, external forces, and equilibration are illustrated in the remainder of the chapter.

3.2.1 Energy approximation errors in 2D mesh interfaces

Though residual forces are a good and intuitive indicator for the performance of a summation rule, the best metric for a summation rule is how well it approximates the energy of a particular configuration, under the condition of QC's kinematic constraints (a mesh, with linear interpolation). The best a summation rule can possibly perform is to calculate energetic quantities at every lattice site and sum them, a strategy that is computationally infeasible. Because of this prohibitive cost, the purpose of a summation rule is to most closely approximate the energy that would be calculated by summing over all lattice sites.

We will perform an investigation to measure how well each summation rule represents the energy of lattice sites in the domain. For any summation rule, the physical volume of the simulation is partitioned and volumes are assigned to each sampling atom as its weight. These volumes, or partitions of space, contain lattice sites. Because of the discrete nature of the lattice site positions and the geometric nature of the volume partition, the sampling atom's weight is not the same as the number of lattice sites within its associated volume, but the two quantities are closely related.

In this example, we will find which sampling atom is representing each lattice site and determine how (in)correct the representation is. We determine the representing sampling atom by finding in which sampling atom's volume a lattice site is found when sampling atom weights are calculated. For example, see Figure 3.2 in which, because of the tessellation, a lattice site is being represented by a sampling atom on the opposite side of its containing element, which has a completely different neighborhood and sees different deformation than the lattice site. As expected, this misrepresen-

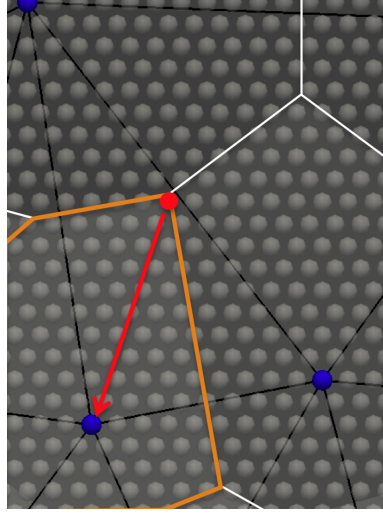


Figure 3.2: Poor sampling atom representation because of tessellation.

tation exhibits itself as incorrect energy and spurious force artifacts, which corrupt equilibrated solutions.

Scenario setup: We start with the (regular) mesh shown in Figure 3.3a, and then we deform it to the state shown in Figure 3.3b and calculate the energy of all lattice sites subject to the QC interpolation, as shown in Figure 3.3c. The representing sampling atom is determined as described above. For illustration, Figures 3.3d and 3.3e draw arrows from each lattice site to the representing sampling atom. Lastly, the (relative) difference between each lattice site’s energy and the representing sampling atom’s energy is computed and plotted, as shown in Figure 3.4. Note that because approximated and exact energies are computed from the same atomic positions, the shown energy errors are merely due to the summation rules and free from approximation errors incurred by the affine QC interpolation.

Results: Figure 3.4 shows energy error per lattice site for an undeformed mesh with sharp interfaces. Of course, in the bottom left corner the error vanishes because full atomistics recovers the exact summation over all lattice sites. With coarse-graining away from the atomistic region, the error increases. We note that because the approximated and exact energies are computed from the same atomic positions, the shown energy errors are merely due to the summation rules and free from approximation errors incurred by the affine QC interpolation.

As expected, each successive summation rule is an improvement. The sampling atom weight tessellation is essentially visible through the errors of the nodal and quadrature rules: the energy of a given sampling atom on a node is not correct for any of its incident elements, so the area of that sampling atom’s Voronoi cell is divided into regions of constant error for each incident element.

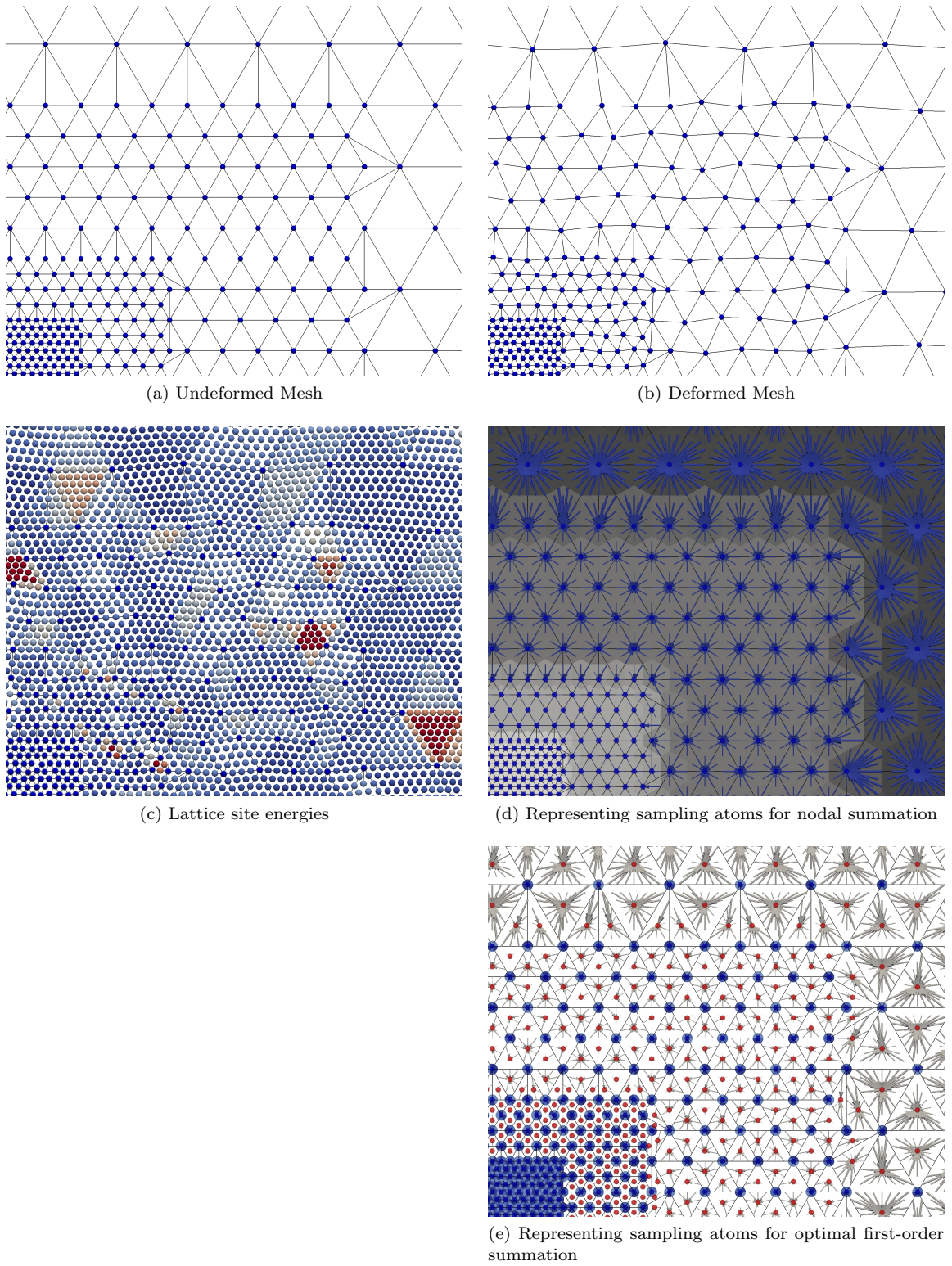


Figure 3.3: The setup of the 2D mesh interface energy approximation example.

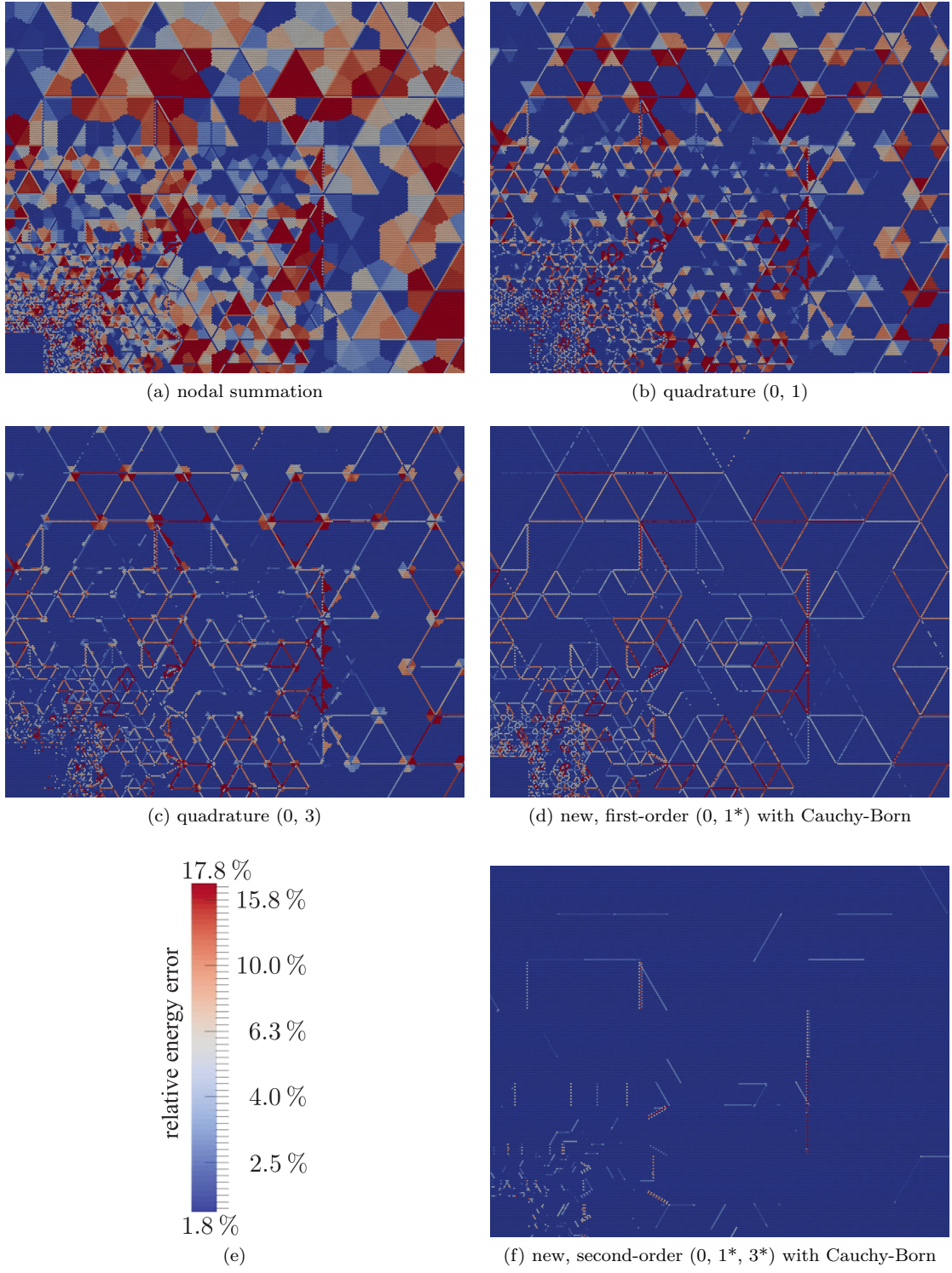


Figure 3.4: Energy error per lattice site for sharp interfaces.

Using single point quadrature introduces large regions inside each element with no error, and more quadrature points increases the size of the region. The first-order new summation rule essentially eliminates the error in the bulk of each element, independent of the element size, and only the errors on the edges remain. With the second-order new summation rule, the error is almost entirely eliminated.

Figure 3.4 shows energy error per lattice site for an undeformed mesh with diffuse interfaces, by which we mean that the undeformed mesh is not regular. Repatom sites are chosen such that the mesh interfaces are randomized, after which the same process (random displacements, lattice site energy calculation, etc.) is followed as for the sharp interfaces. The results are qualitatively the same as for sharp interfaces, except that quadrature rules perform worse because the Voronoi cells do not line up as well as they did before. The energy approximation errors show exactly the same trends discussed above for two dimensions; for concision and because of visualization difficulties we omit them here.

To quantify the total error in Figures 3.4 and 3.5, Figure 3.6 shows the average energy approximation error per atom for the different summation rules. Repatoms are displaced randomly by a fraction of the distance to their nearest neighbor repatom (this fraction is shown as the normalized random displacement). Both graphics use the same legend to identify the various summation rules; errors are shown on a logarithmic scale. In particular, results of Figures 3.4 and 3.5 were obtained with a normalized random displacement of 0.2 and are included in Figure 3.6. Using the same meshes of Figure 3.4 with sharp and diffuse interfaces, we plot the average energy error of all lattice sites in the crystal with increasing (randomized) displacements (normalized by the distance to their nearest neighbor repatom) of all repatoms from their equilibrium positions. Error bars represent the standard deviation of 100 runs of each data point.

Nodal summation is markedly worse than any other rule, and the quadrature and cluster rules are fairly clumped. The first-order optimal rule outperforms all other rules for substantially lower cost, but the second-order optimal rule is significantly better than first-order, especially in the undeformed configuration. While the new schemes performed well in the residual force experiments above, residual forces only apply in the undeformed configuration and it is not clear that the superiority holds in deformed meshes. However, these results here demonstrate that the new scheme also performs the best of the summation rules in deformed meshes, resulting in lower energy errors and reduced spurious forces.

Cost: A key difference between the summation rules is their cost, something we have not yet examined. The cost of a particular sampling atom is a function of how many neighbors it has. For this investigation, all sampling atoms have similar numbers of neighbors, so the number of sampling atoms used is a good proxy for the cost of the summation rule.

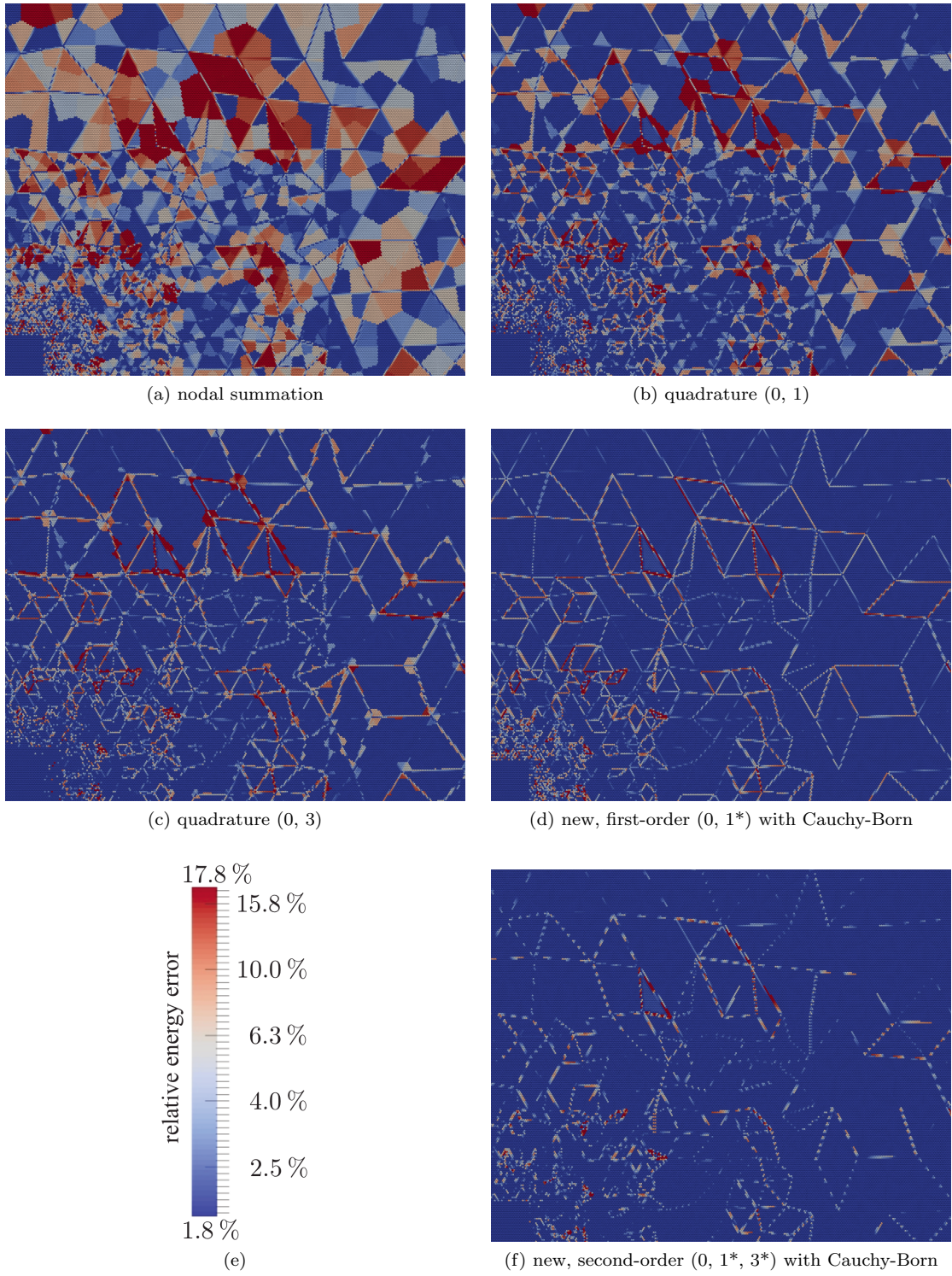


Figure 3.5: Energy error per lattice site for diffuse interfaces.

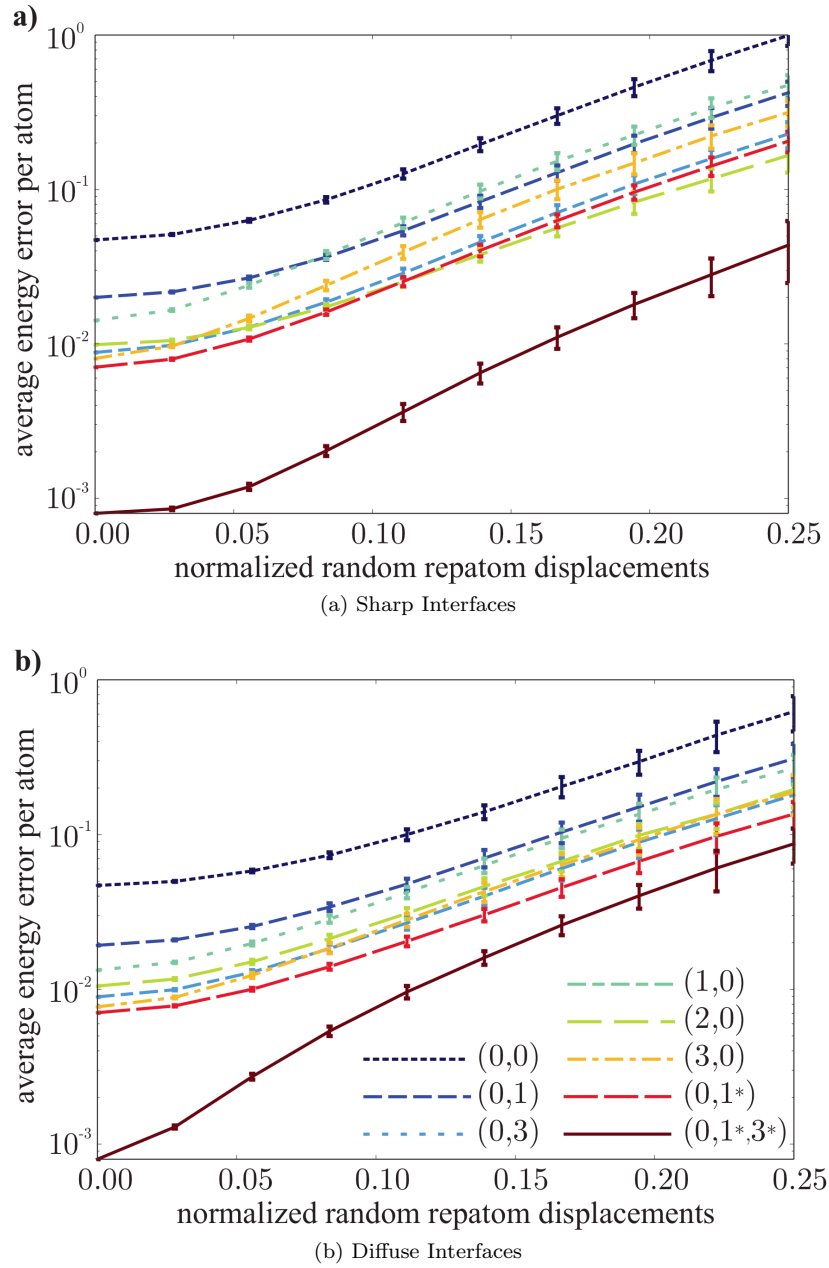


Figure 3.6: Average relative energy error per atom in the 2D QC meshes of Figures 3.4 and 3.5 containing a) sharp and b) diffuse interfaces.

Figure 3.7 shows a scatter plot of the total energy error in the undeformed configuration versus number of sampling atoms used. Each summation rule has three crosses, one for sharp interfaces and two for increasingly-more diffuse interfaces. Nodal summation is the cheapest rule and has the worst error. Clusters provide lower error, but using more shells than one greatly increases the cost without greatly decreasing error. Using one- and three-point quadrature improves the error for increasing

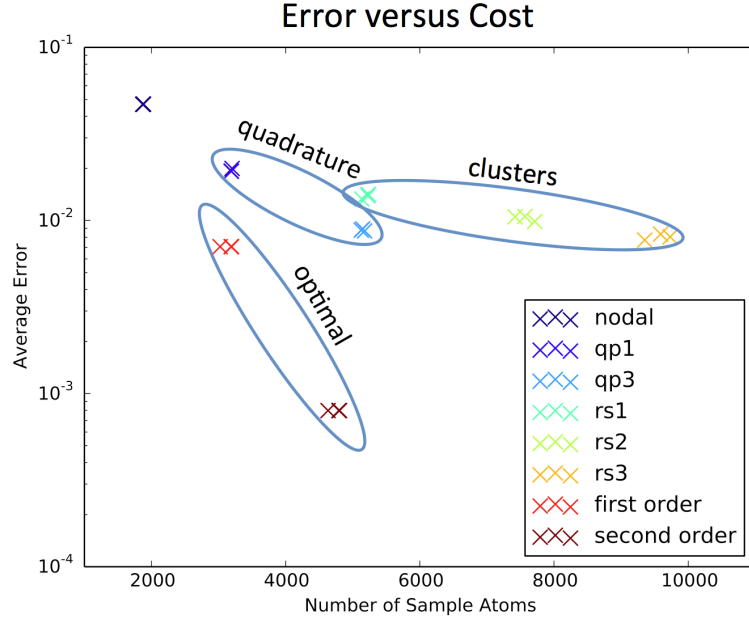


Figure 3.7: Interface energy approximation error versus cost for various summation rules.

cost, such that using three points costs about the same as one-shell clusters, but with lower error.

The new first-order optimal summation rule provides a half order of magnitude lower error for the same cost as one-point quadrature. Second-order optimal performs phenomenally, achieving more than an order of magnitude lower error than three-point quadrature, for slightly less cost. The new summation rules more accurately approximate the total energy of the system for lower cost than any existing rule and perform in a class of their own.

3.2.2 Residual forces in 2D mesh interfaces

Because “ghost”, or residual forces are used so often to criticize the QC method, their presence and magnitude are natural measures for summation rules. We first examine residual forces in two dimensions, after which we will investigate them in three dimensions. The scenario will be a single-crystal Copper simulation modeled by the extended Finnis-Sinclair potential of Dai et al. [2006], with a domain that is rapidly coarse-grained from full atomistic resolution in four steps with coarsening factors of 2, as shown in Figure 3.8a. Residual forces would traditionally be observed at each mesh interface, but not within the regions of uniform coarsening.

Figure 3.9 shows the residual forces for the nodal summation rule (0,0), cluster summation rule (3,0) (each 3-shell cluster contains 54 lattice sites), classical quadrature summation rules with one and three sampling atoms per element, and the new summation rules of first and second order. In addition, we compare results obtained with and without using the Cauchy-Born rule for all non-repatom sampling atoms. It becomes apparent that the new schemes perform much better than the

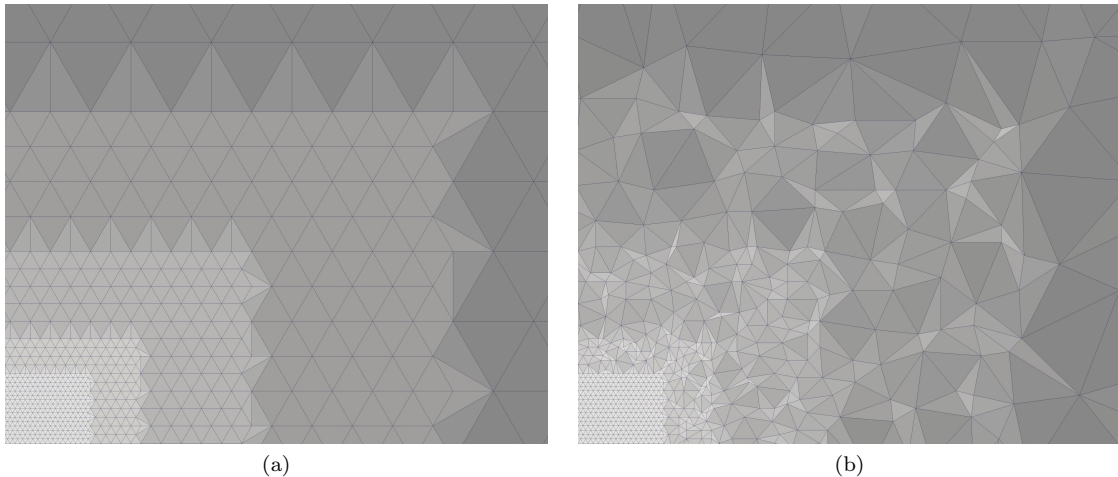


Figure 3.8: Meshes for the sharp (a) and diffuse (b) mesh interface examples illustrated in Figures 3.9 and 3.10.

previously-existing schemes, while being much more efficient than, for example, the cluster rule due to the small number of sampling atoms. As can be expected, the second-order scheme is superior to the first-order scheme, producing smaller residual forces and confining these to only those repatoms adjacent to the full atomistic region.

Using the Cauchy-Born rule The application of the Cauchy-Born rule for inner-element sampling atoms appears to be beneficial. While this does not affect the energies and forces in atomistic or in sufficiently large elements, it does affect those small elements in between. The benefits of the Cauchy-Born rule can be explained by the restoration of symmetry in each element by application of the Cauchy-Born rule: without the Cauchy-Born rule, every quadrature-type sampling atom must be placed on an actual lattice site to define its atomic neighborhood. In small elements, this sampling atom may be placed significantly far away from the element’s barycenter, depending on the element size and geometry. The resulting (non-affinely deforming) neighborhoods produce forces in the undeformed configuration (and introduces force artifacts in the undeformed configuration). To allow for a fair comparison, we have also included the traditional quadrature rule (0,3) with sampling atoms not tied to actual lattice sites; see Figure 3.9e. Even in this case, the superior performance of the new schemes is apparent.

Similarly, Figure 3.10 compares residual forces arising in a non-uniform QC mesh with diffuse mesh interfaces (mesh shown in Figure 3.8). The diffuse mesh interface scenario is much more difficult for summation rules; residual forces would traditionally be observed at throughout the domain because there are no regions of uniform coarsening.

Despite the randomized repatom locations, sizes, and aspect ratios, the new summation rules

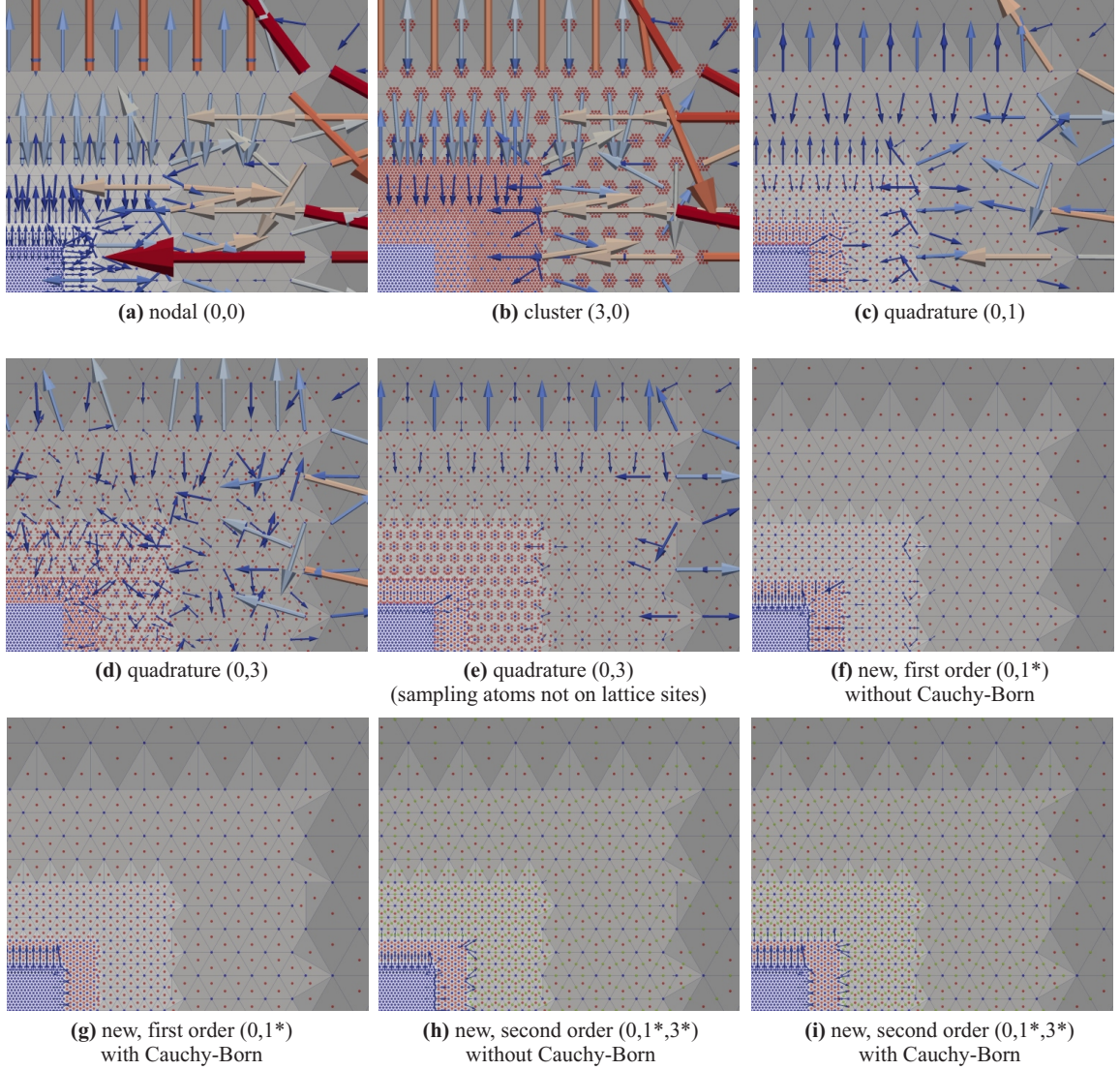


Figure 3.9: Residual forces at sharp mesh interfaces in two dimensions for various summation rules (coarsening factor of 2 at each interface), using the extended Finnis-Sinclair potential of [Dai et al., 2006] for Cu. The color code of arrows reflects the magnitudes of residual forces (in arbitrary units); small dots represent repeatoms (blue) and additional sampling atoms (red/green). The mesh is shown in Figure 3.8a.

perform significantly better, showing residual forces of much smaller magnitude and at fewer locations. Here, the second-order scheme surprisingly shows larger forces which arise from the strong geometric irregularities of the randomized mesh. Overall, the new summation rules confine residual forces to a small band of element sizes right above full atomistics and below elements exceeding $2n_b - 1$ lattice sites per side, as can be expected from our analysis. Thus, the ghost forces observed in the traditional local/nonlocal QC are effectively smeared over a regime of element sizes (rather than being localized at the sharp interface between atomistic and coarse-grained regions). Yet the new scheme allows us to treat the entire domain in exactly the same manner without explicitly

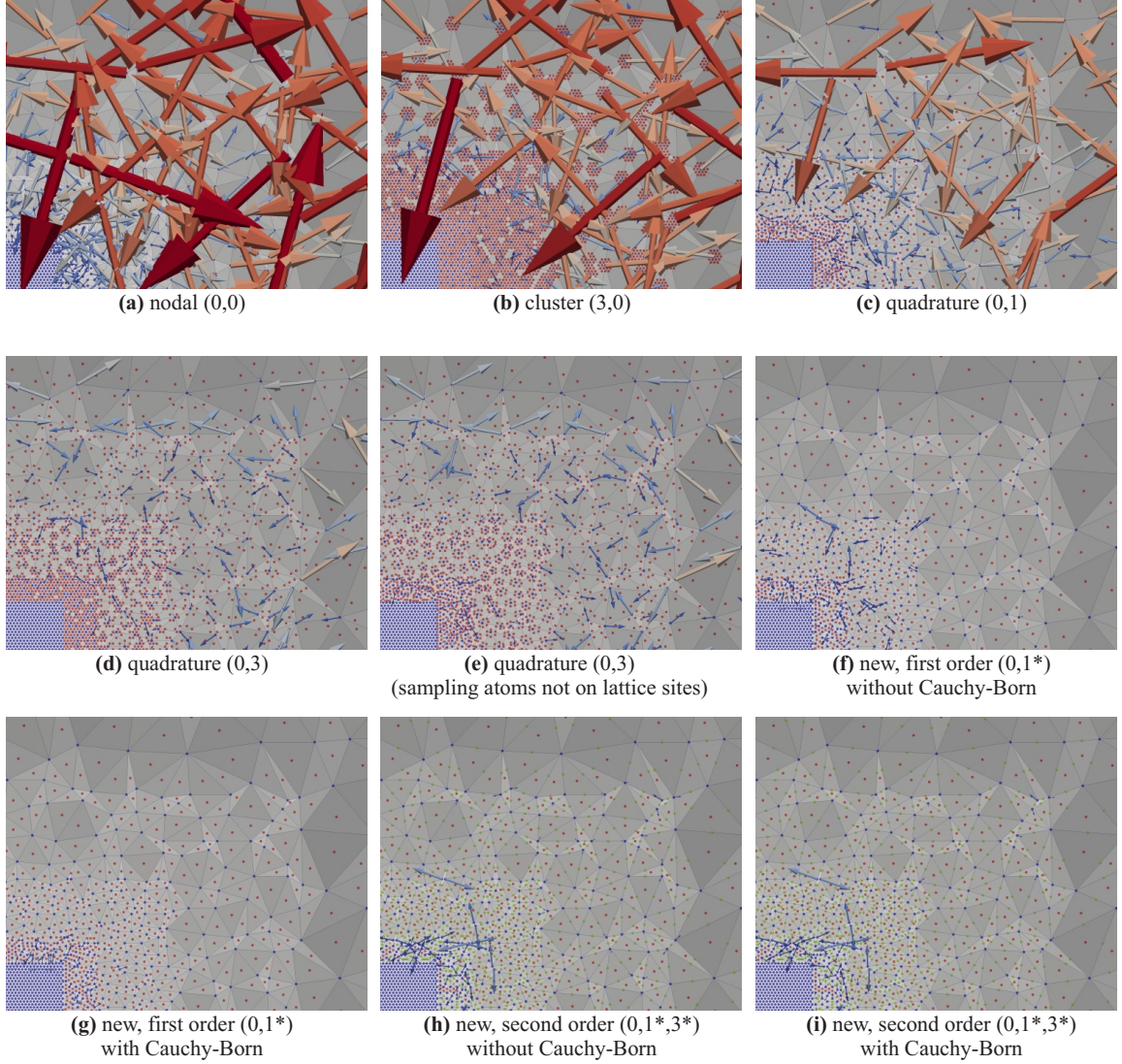


Figure 3.10: Residual forces at diffuse mesh interfaces in two dimensions for various summation rules (repatom positions of Figure 3.9 have been perturbed randomly), using the extended Finnis-Sinclair potential of [Dai et al., 2006] for Cu; the color code reflects residual force magnitudes. The mesh is shown in Figure 3.8b.

differentiating between local and nonlocal repatoms or elements.

3.2.3 Residual forces in 3D mesh interfaces

Figures 3.11 and 3.12 demonstrate residual forces in non-uniform QC meshes with sharp and diffuse interfaces in three dimensions for the same summation rules discussed above. The new summation rules use the Cauchy-Born rule for non-repatom sampling atoms; the color code reflects residual force magnitudes (in arbitrary units). Here, nodal cluster and quadrature rules reduce residual force artifacts more effectively than in two dimensions, yet the new summation rules are again shown

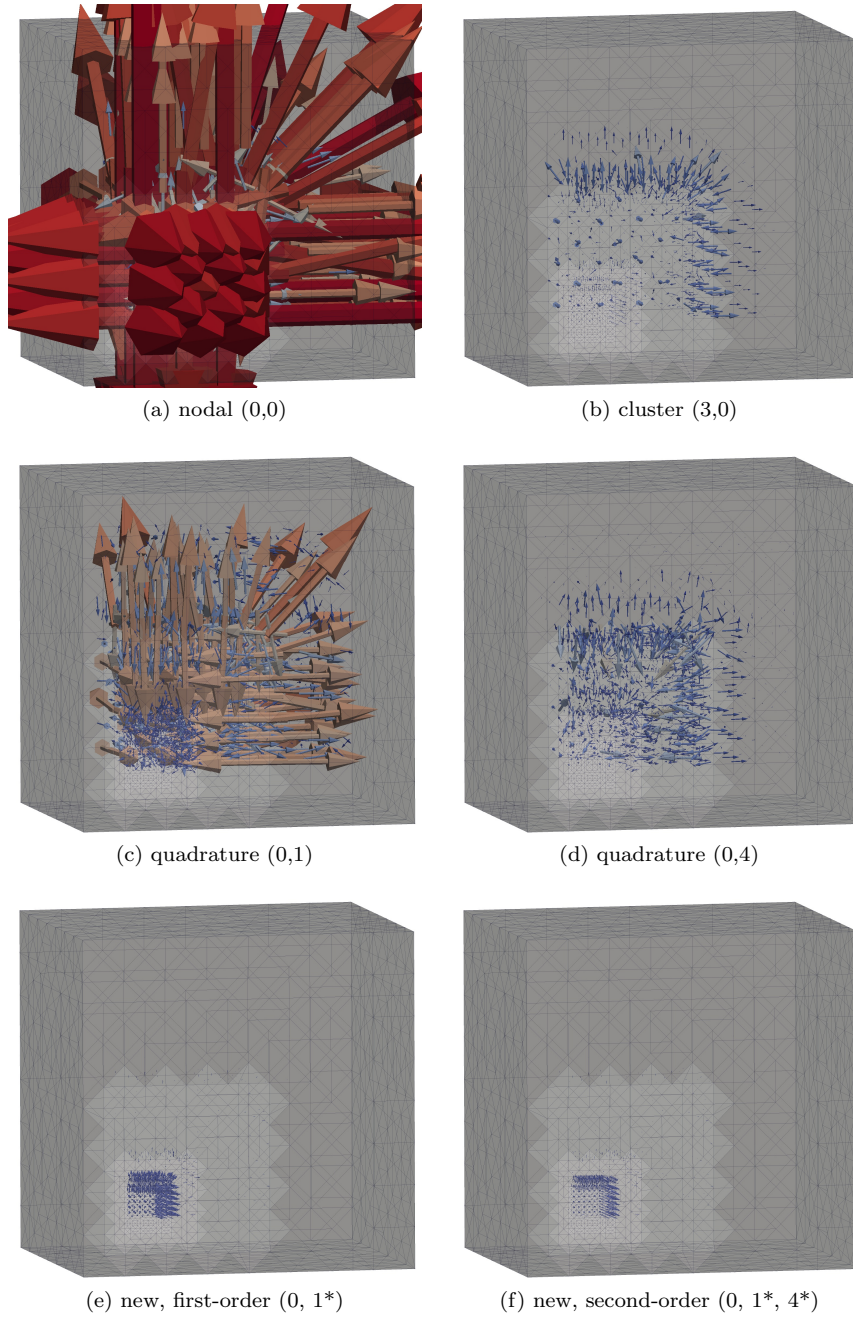


Figure 3.11: Residual forces in non-uniform QC meshes with **sharp** interfaces.

to be superior to all others, despite the geometric shortcoming discussed in Section 2.4. Both the new first and second-order schemes reduce force artifacts to small elements right above atomistic resolution, as can be expected from our analysis.

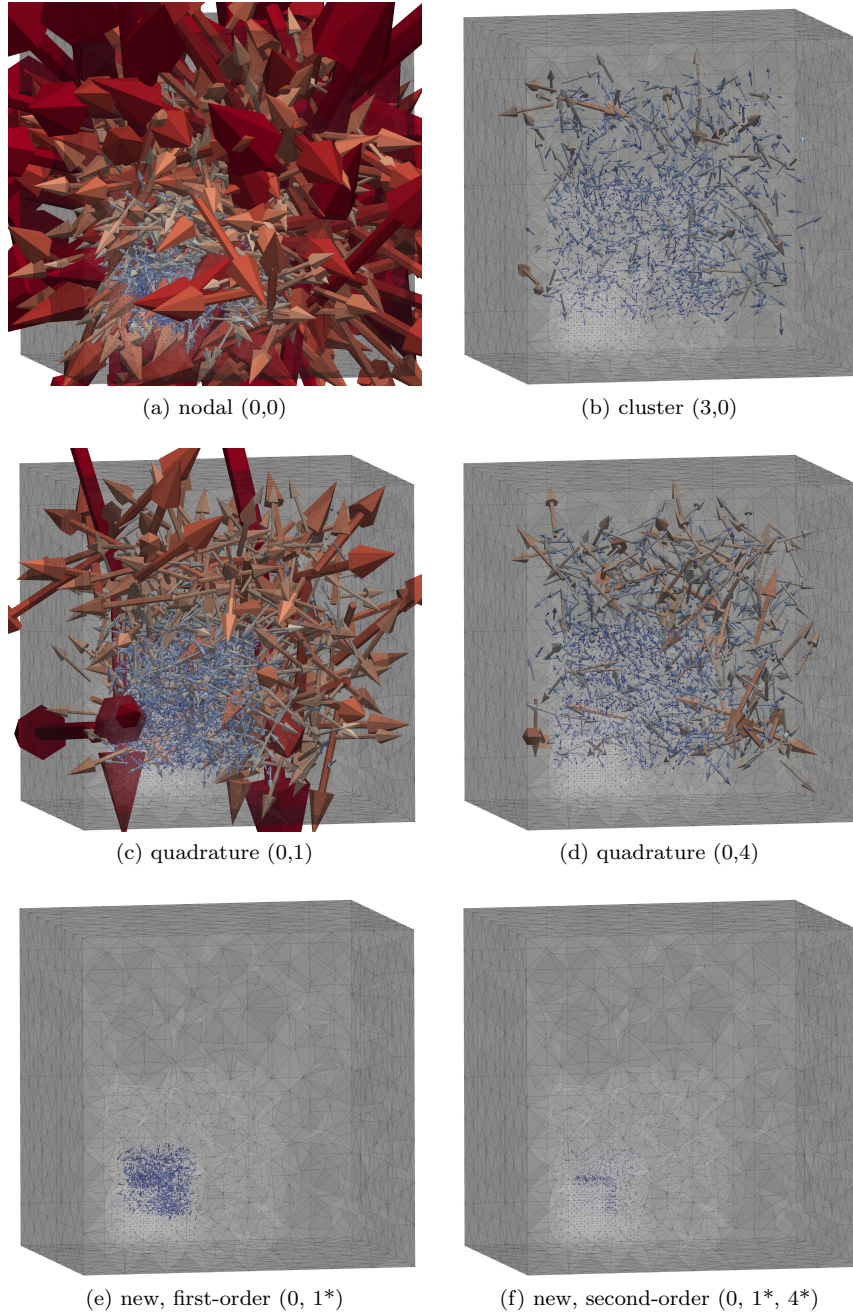


Figure 3.12: Residual forces in non-uniform QC meshes with **diffuse** interfaces.

3.2.4 Summary of examinations of isolated configurations

In this section, we have examined the performance of the various summation rules via energy approximation errors and residual force artifacts. The results we observe here hold true for the remainder of the examinations of the this work: nodal summation performs very poorly, clusters improve accuracy marginally while greatly increasing cost, and quadrature performs well (especially with

multiple quadrature points). However, the novel summation rules greatly outperform the legacy rules, exhibiting significantly reduced energy approximation errors for all element sizes with vanishing residual force artifacts in the small- and large-element limits, in uniform *and* non-uniform QC meshes. The remaining residual forces are confined to a small range of element sizes between full atomistics and sufficiently large for non-overlapping repatom spheres of interaction. This extends the ghost forces of the local/nonlocal QC method into a fully-nonlocal framework with improved model adaptability.

3.3 Elastic benchmark simulations in three dimensions

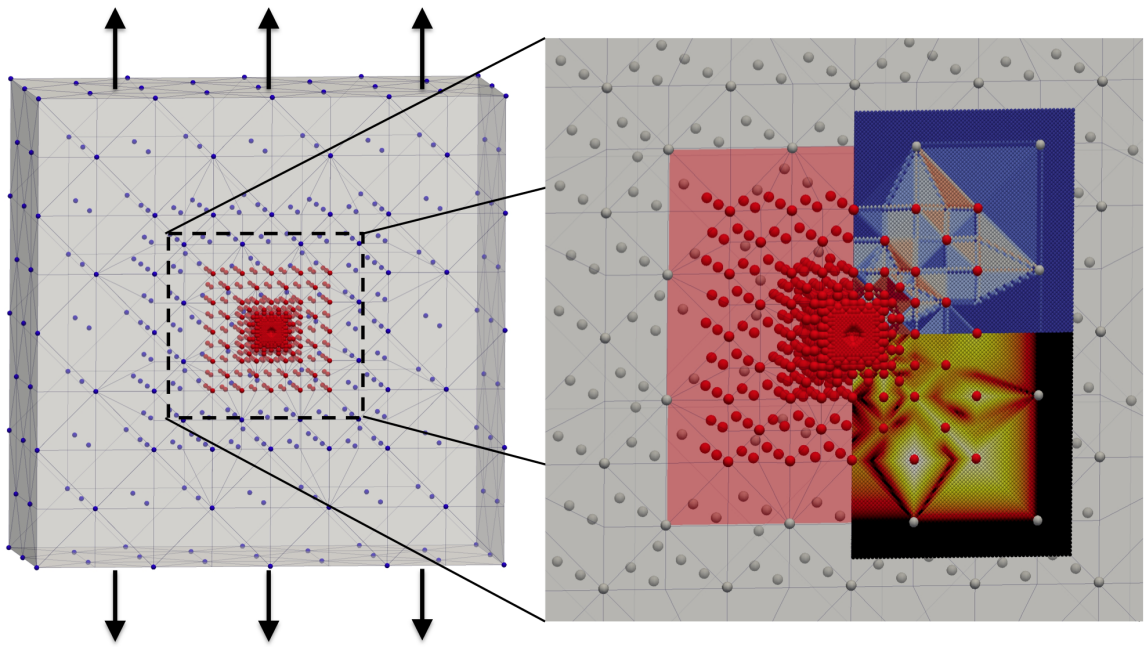
The QC method is primarily used to approximate long-range elastic fields. In the immediate vicinity of defects, full atomistic refinement turns our nonlocal QC scheme into lattice statics with no approximation errors or residual forces. Therefore, to analyze the performance of the QC formulation in this section, we concentrate on scenarios where the repatom spacing is larger than the atomistic limit.

Moreover, once plastic microstructure starts to form, simulation results show statistical deviations (even fully-atomistic results are not exactly reproducible when using any form of parallelism in general, due to floating point arithmetic). Therefore, our quantitative benchmark examples in this section will concentrate on finite yet elastic deformation. Results beyond the onset of plasticity will be compared qualitatively to solutions obtained from lattice statics in Section 3.4.

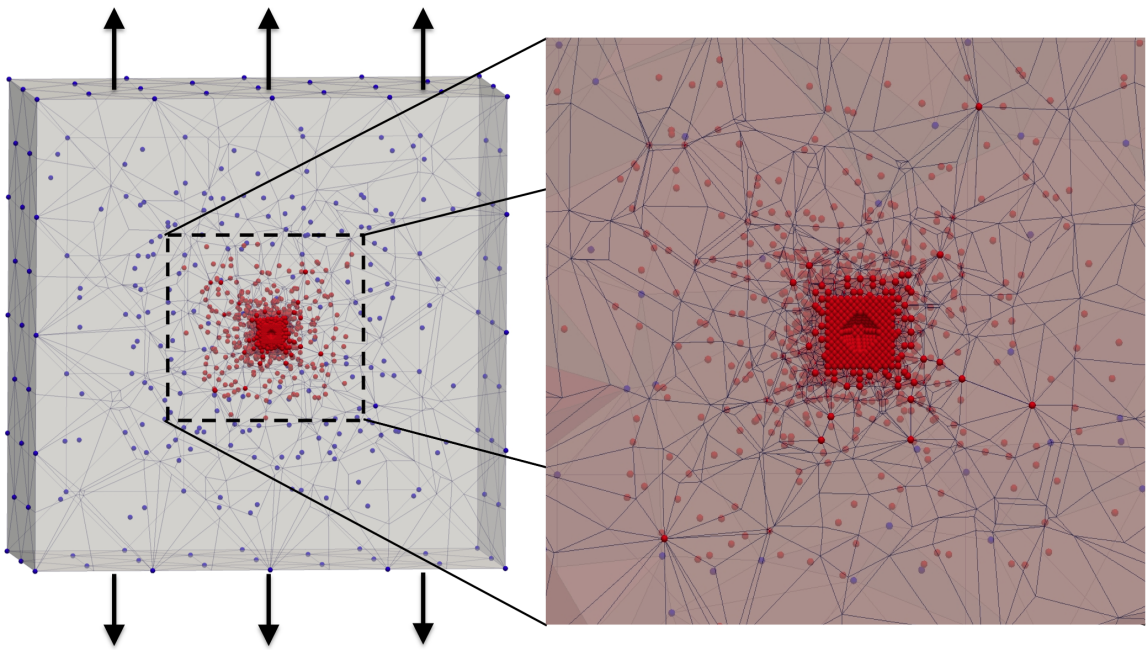
3.3.1 Local position and energy errors in a nano-void simulation

Most commonly, QC simulations are performed on domains with relatively small atomistic regions and a surrounding coarsened mesh. Though the mesh may be coarsened aggressively, gradual coarse-graining is favored to reduce approximation errors due to summation rules and interpolation schemes (and is automatically achieved by intelligent model adaptation). Meshes that do not change over the course of the simulation (if model adaptation is disabled) often contain sharp mesh interfaces, while meshes which are changed because of model adaptation often contain diffuse and randomized mesh interfaces.

We start our examples by assessing the performance of the various summation rules in such meshes, where we study long-range inhomogeneous elastic deformation. Note that here and in some of the following examples deformation is enforced through affine boundary conditions applied to a buffer region around the actual simulation core. Since atomic potentials generally require more than next-to-nearest neighbor interactions, such a buffer region (a common technique in MD) is required to avoid parasitic surface effects by providing all required neighbors to the core region. This, of course, is only an approximation to impose a desired average remote loading. Since our



(a) Sharp, structured mesh interfaces



(b) Diffuse mesh interfaces

Figure 3.13: Examples of single-crystalline cubes of pure copper with a nano-void.

QC formulation allows us to, in principle, coarse-grain aggressively to arbitrary sample sizes, it is well possible to apply those boundary conditions arbitrarily far away from the simulation core. This avoids spurious cell-size effects that can arise in atomistic calculations; see for example [Szajewski

and Curtin, 2015].

The scenario is a Cu single-crystal with a single spherical void (diameter 5 nm). The void leads to an inhomogeneous stress distribution with concentrations in the vicinity of the void. To properly resolve the void, we assign atomistic resolution to its immediate vicinity and rapidly coarse-grain away from the void in a total of six steps with coarsening ratios of roughly 2, as shown in Figure 3.13. Affine boundary conditions of uniaxial extension are applied to all blue repatoms, while the core regions defined by all red repatoms are relaxed to equilibrium. The zoomed images show the relaxed QC regions; Figure 3.13a includes example results of energy and displacement errors. The sample is loaded at its top and bottom in uniaxial extension, using affine boundary conditions to remove surface effects and mimic an infinite single crystal. The cube is stretched vertically in the $[100]$ direction and relaxed to equilibrium. After equilibrium is achieved, the relaxed positions of all lattice sites are computed from the QC model using its affine interpolation rules. We also compute all exact atomic positions from a geometrically-equivalent lattice statics model with the same boundary conditions, and proceed to compare the two results by energy and displacement errors.

The system contains $67.1 \cdot 10^6$ lattice sites whose interactions are modeled by two different QC meshes: one mesh containing sharp interfaces (comprising about 7,140 repatoms and 41,800 elements) and one mesh that is seamlessly coarse-grained by randomized, diffuse interfaces (roughly 7,130 repatoms and 42,500 elements). The mesh with sharp interfaces is representative of one made by hand in a simulation without model adaptation, and the mesh with diffuse interfaces is representative of the result of automatic model adaptation. In both cases, the number of degrees of freedom has been reduced to about 0.01%.

Microstructure evolution is certainly an application of great interest for the QC method, but we deliberately limit our discussion to elastic loading in the present example for various reasons:

- a) To demonstrate the approximation power of long-range elastic fields in a radically coarse-grained QC model.
- b) QC is essential in approximating long-range elastic fields while atomistic defects require locally full atomistic resolution (capabilities to study complex microstructural mechanisms will be shown in Section 3.4).
- c) The formation of inelastic defects automatically introduces randomness of solutions, which precludes one-to-one comparisons of lattice site positions to solutions obtained via lattice statics.

Figure 3.14 illustrates the displacement errors and relative energy errors of all lattice sites compared against the exact atomistic solution for various summation rules applied to the sharp-interface QC mesh. The single-crystalline Cu nanocube with a spherical void undergoes uniaxial extension in the vertical $[100]$ direction. Errors are shown on logarithmic scales, and displacement errors are shown in Angstroms. Each graphic results from a particular summation rule and illustrates the

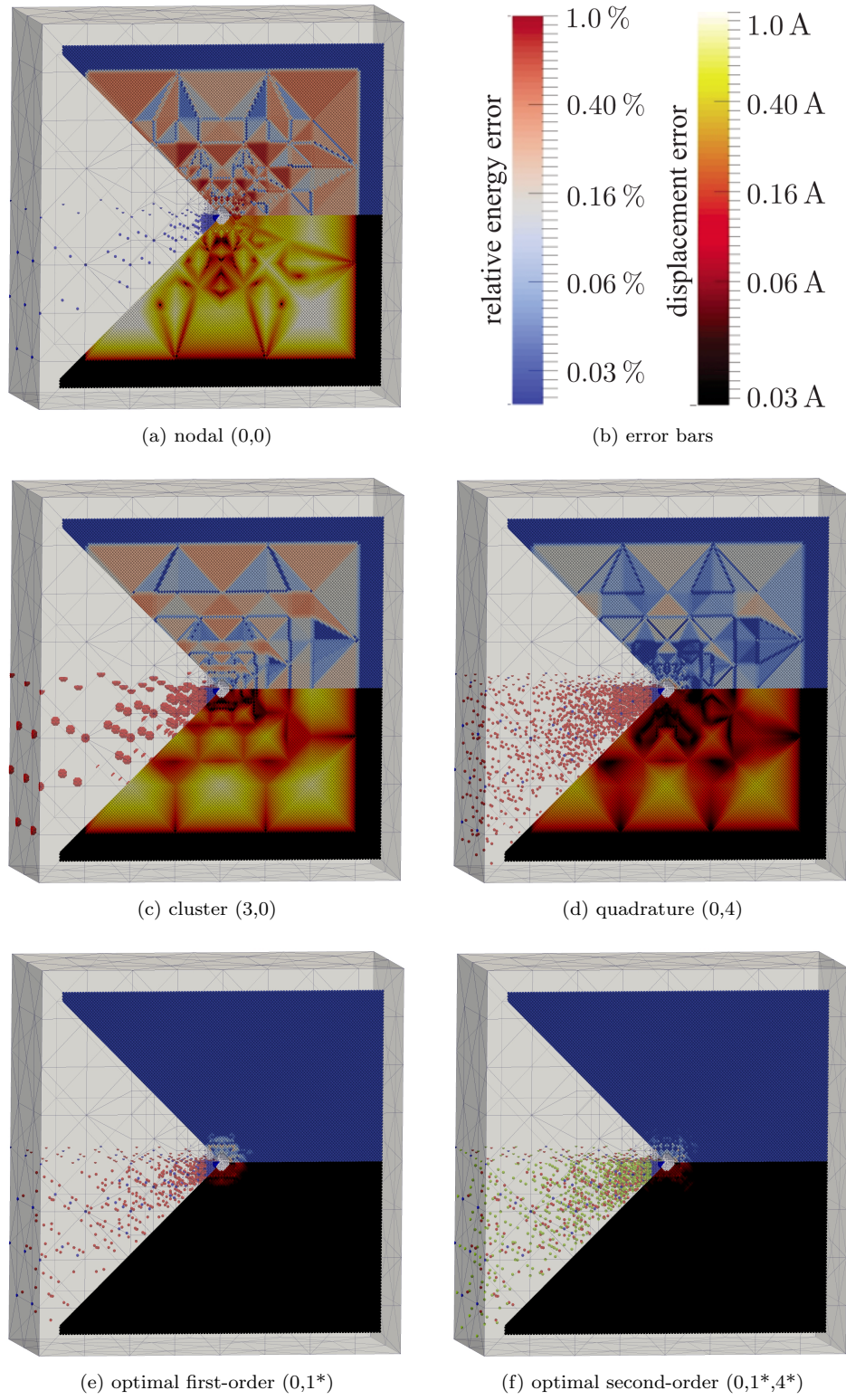


Figure 3.14: Single-crystalline Cu nanocube with a spherical void, with *sharp* interfaces.

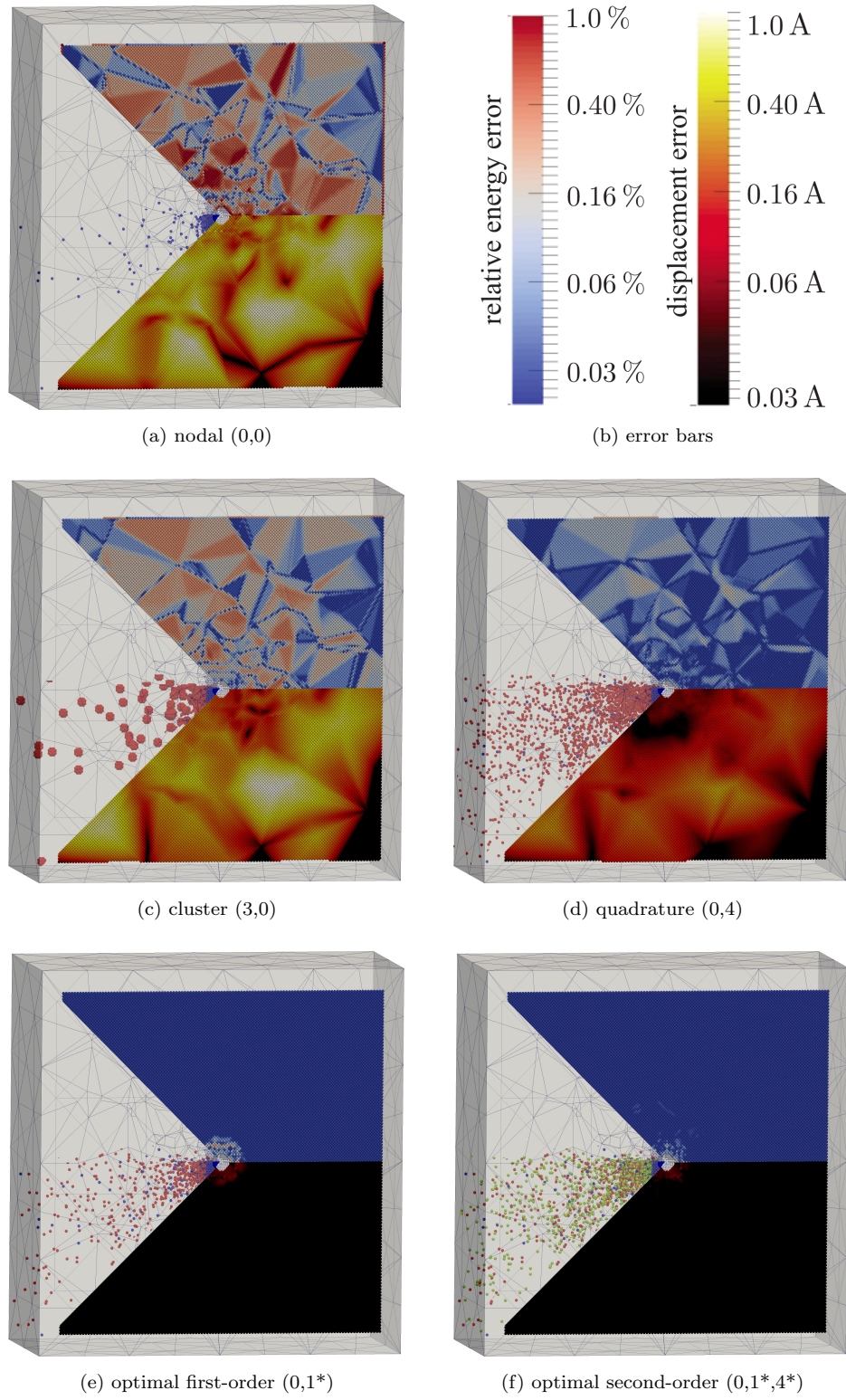


Figure 3.15: Single-crystalline Cu nanocube with a spherical void, with *diffuse* interfaces.

distribution of repeatoms and sampling atoms (left), the displacement errors of all lattice sites compared against full atomistics (bottom), and the corresponding relative energy errors of all lattice sites (top). For reference with regards to displacement error, the lattice spacing in Cu is approximately 3.61 Å.

For the nodal and cluster summation rules, displacement errors show significant values throughout all coarse-grained regions. The same trends as in previous examples are observed in both energy and displacement errors: the introduction of nodal clusters reduces the error of the nodal summation rule (0,0) with increasing cluster size (see for example (3,0)), and quadrature summation rules reduce errors below those of cluster rules. Even the lowest-quadrature rule (0,1) produces smaller errors than the computationally much more expensive (3,0) rule in larger elements.

In regions with close-to-atomistic resolution, cluster rules and quadrature rule (0,4) produce minimal errors because essentially all lattice sites are turned into sampling atoms, greatly increasing the size of atomistic regions and resulting in poor efficiency. The new summation rules (whose efficiency is comparable to (0,1) and (0,4) quadrature for the first- and second-order schemes, respectively) reveal by far the lowest errors overall with only marginal differences between the first and second-order schemes.

We note that, because this scenario is an integral test with boundary conditions and equilibration, the vanishing displacement errors are not only an indication of the excellent performance of the summation rules but also confirm the general applicability of the QC approximation to represent long-range elastic fields. Also, it is important to realize that errors of the traditional summation rules do not only arise in the coarse-grained regions but propagate into the full atomistic core as well, primarily due to spurious force artifacts. Figure 3.15 shows the equivalent displacement and energy errors for the diffuse QC mesh. As can be expected, errors are larger than in the case of sharp mesh interfaces but reproduce the same trends. Note that in both cases, the second-order scheme performs better than the first-order rule.

potential	cut-off distance	lattice spacing	radius r_w	potential reference
Johnson	3.249000 Å	3.609966 Å	1.7 Å	Johnson [1989]
ext. Finnis-Sinclair	4.320000 Å	3.609966 Å	2.4 Å	Dai et al. [2006]
Voter	4.961000 Å	3.614999 Å	2.5 Å	Mishin et al. [2001]
Mishin	5.506786 Å	3.615000 Å	2.4 Å	Mishin et al. [2001]
Mendelev	6.000000 Å	3.635871 Å	2.1 Å	Mendelev and King [2013]

Table 3.1: Overview of Cu interatomic potentials used for a comparison of approximation errors, along with their tabulated cut-off distances as well as the lattice spacing obtained from energy minimization in a perfect infinite crystal lattice (rounded to six digits). The optimal effective interaction radii r_w for the definition of sampling atom weights for the five potentials are also included.

EAM potentials: We would like to demonstrate that the new scheme performs equally well for all EAM potentials and is not tied to the specific interatomic potentials used in our examples. To this end, we compare five atomic potentials which greatly differ by their radii of interaction (see Table 3.1). In each case, we have determined the effective interaction radius r_w (required to determine the weights of sampling atoms) by numerical experimenting. This radius can be found a priori, which is required only once per potential.

Specifically, we have repeated the void-expansion simulations of Figure 3.15 for various effective interaction radii r_w and have chosen those values as optimal that produce the least total position error of all lattice sites in the sample. Of course, this does not guarantee that the determined radii are optimal for arbitrary simulation scenarios. However, the chosen example involves a wide range of element sizes in an unstructured mesh and was selected as a representative scenario. Though there is clearly room for improvement here, even this rather ad-hoc choice performs orders of magnitude better than the other summation rules. A more detailed analysis of optimal values for the effective interaction radii in particular in the presence of free surfaces can be found in Section 3.5.

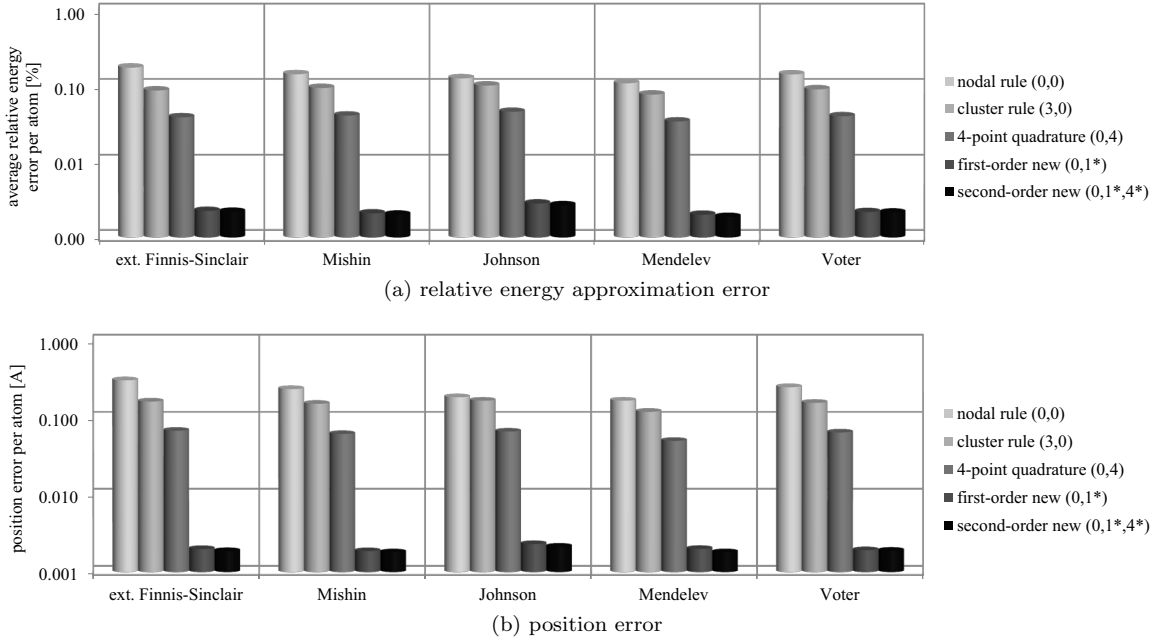


Figure 3.16: Comparison of errors of all lattice sites for the example of Figure 3.13 (a nanocube with a void using a QC mesh with sharp interfaces and undergoing uniaxial extension) for five different summation rules and the interatomic potentials of Table 3.1.

The graphs in Figure 3.16 summarize the average relative energy and position errors, respectively, of all lattice sites contained in the cube with a nanovoid of Figure 3.13b for five different summation rules and for all potentials listed in Table 3.1. The general qualitative conclusions drawn above hold for all five potentials: the new first- and second-order summation rules produce significantly smaller

errors compared to nodal, cluster and traditional quadrature rules (note the logarithmic scales in both graphics). This verifies that the summation rule performance is not tied to a particular choice of interatomic potential.

We note that even though our numerical examples used potentials of the Embedded Atom Method, the presented methods are sufficiently general to apply to, in principle, any distance-dependent interatomic potential. The new summation rules show their beneficial effects when applied to any centrosymmetric lattice.

3.3.1.1 Effect of residual force removal

In the traditional QC method as well as in general atomistic-continuum coupling schemes, force artifacts appear both as residual forces (i.e., in the undeformed configuration) and as spurious forces (i.e., errors in repatom forces in the deformed configuration). It is easy to determine residual forces by simply computing forces in the undeformed configuration, though one must be careful to filter out any forces that come from free surfaces. It is extremely difficult to determine spurious forces. Various corrections have been introduced to remove such force artifacts, in particular in coarse-graining schemes that explicitly differentiate between atomistic and coarse-grained regions. There, one commonly speaks of *ghost force corrections*; see for example [Shenoy et al., 1999b, Ortner and Zhang, 2014].

A misconception is that the removal of residual forces by appropriate dead loads on repatoms (determined in the undeformed, initial state) is sufficient to significantly reduce force artifacts in the deformed state in a fully-nonlocal QC scheme, and we can use the same nano-void example to illustrate that this is indeed not the case. To this end, we take the example of Figure 3.13b with different summation rules; we determine the residual forces in the undeformed state and remove those by (constant) dead loads on all repatoms. Then, we apply uniaxial extension and relax the sample in the presence of the corrective dead loads. Of course, the undeformed configuration will now be reproduced exactly but the deformed state displays the errors shown in Figure 3.17. Apparently, the effect of the dead load correction (see the bottom row) is marginal and for some of the summation rules even produces larger errors than without correction at sufficient levels of deformation.

This example shows that attempting to correct *spurious* forces by simply removing *residual* forces as dead loads is insufficient. Note that correction methods have been introduced in local/nonlocal QC, cf. [Shenoy et al., 1999b], which effectively reduce ghost forces at the local/nonlocal interfaces by recomputing corrective dead loads after each iteration step. However, this affects the conservative energy-based nature of the QC scheme, it requires special procedures for repatoms near the local/nonlocal interface, and it requires the notion of an interface (which does not exist in the fully-nonlocal formulation here). We maintain that the best way to effectively remove both residual and spurious force artifacts is by minimizing energy approximation errors via optimal summation rules

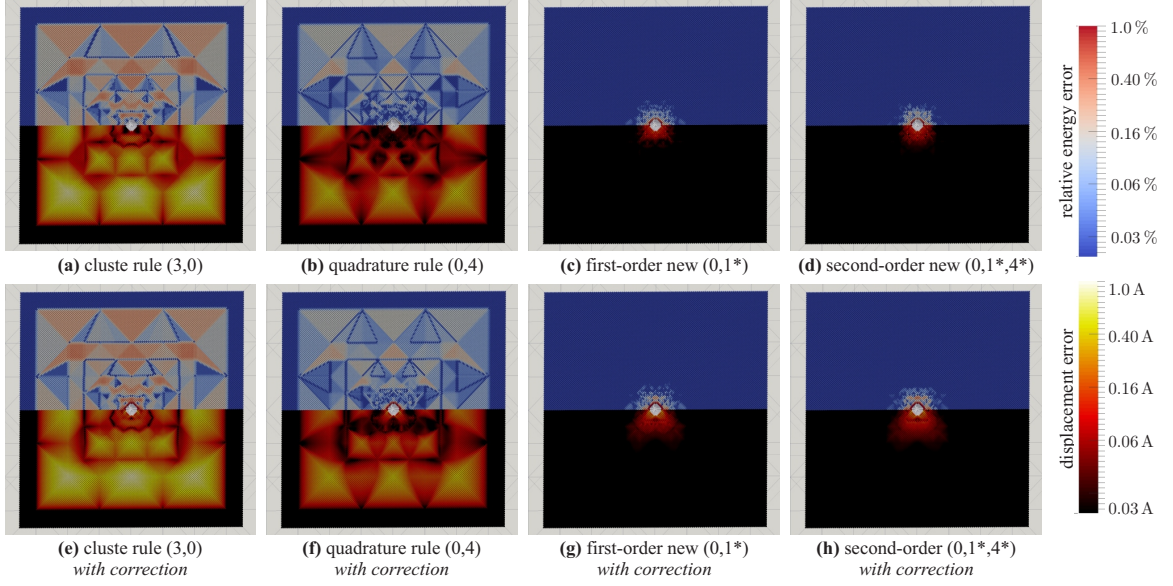


Figure 3.17: Influence of removing residual forces by dead loads by comparing errors produced by four different summation rules (a-d) without and (e-h) with removed residual forces; top halves of all graphics illustrate relative energy errors per atom, and bottom halves show displacement errors of all lattice sites for the sharp-interface scenario of Figure 3.13b.

such as those investigated here.

3.3.2 Elastic constants in randomly coarse-grained copper single crystals

In this section, we shall examine the ability of the various summation rules to reproduce known elastic constants for varying repatom density. Some details have been omitted; a full treatment can be found in Amelang et al. [2015]. Dr. Gabriela Venturini performed much of this work in collaboration.

Scenario setup: To assess the total error introduced by the various summation rules, we compute elastic constants in three-dimensional QC representations of an infinite fcc copper single-crystal modeled by the extended Finnis-Sinclair potential of Dai et al. [2006]. Affine boundary conditions are applied remotely to a large coarse-grained crystalline ensemble whose repatom locations are chosen randomly from the entire set of lattice sites. This qualifies as a worst-case scenario of severely non-uniform meshes of poorest quality. We define the (repatom) *density* of a QC mesh as the ratio of the number of repatoms to the total number of lattice sites contained in the sample.

For each chosen value of density, the specific distribution of the randomly-chosen repatoms affects the results. Therefore, multiple realizations of each system size and summation rule choice are required to reliably determine mean relative errors and standard deviations (all errors are calculated against (theoretically) exact elastic constants obtained from a fully-atomistic description).

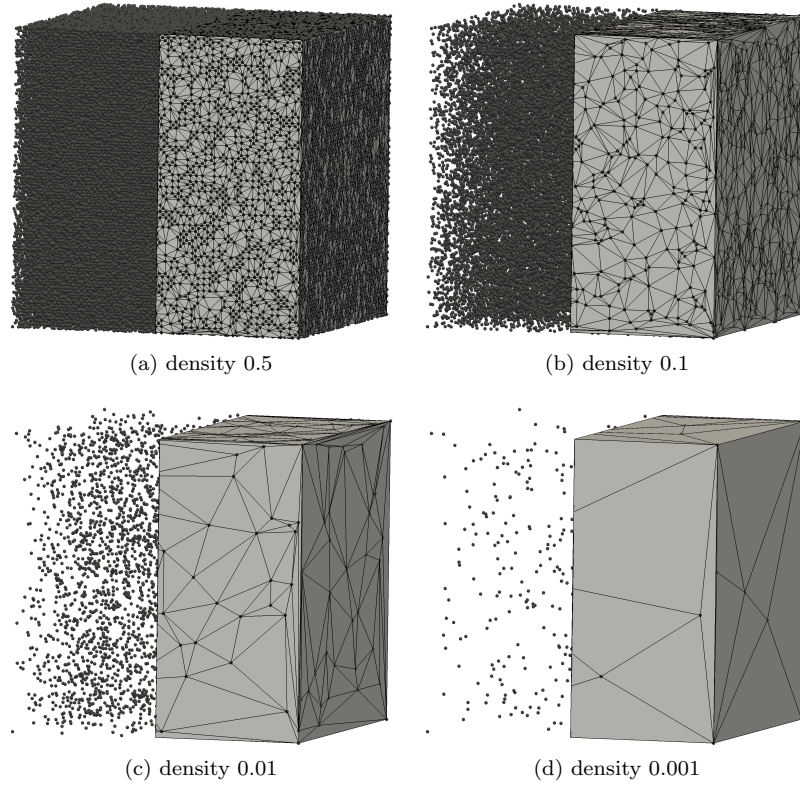


Figure 3.18: Example repatom distributions in randomly coarse-grained crystals for four different repatom densities: a) 50%, b) 10%, c) 1%, and d) 0.1%. The right half of each cube illustrates tetrahedral elements used for interpolation and shown on the surface of the cube, the left half shows repatom locations (grey dots represent repatoms, solid black lines are element edges).

Figure 3.18 illustrates example distributions of repatoms for four different densities along with the associated QC meshes. In order to guarantee a minimum number of repatoms inside the sample (especially for densities below 1%) system sizes are chosen to increase with decreasing density.

Any reasonable summation rule must reproduce the exact results in uniformly coarse-grained samples (and all summation rules studied here do so within numerical tolerances). As representative examples, we investigate the four more traditional summation rules shown in Figure 3.19 along with the new summation rule of first order. It is important to note that the reported errors of the elastic constants stem from two main sources (besides numerical noise). On the one hand, the QC approximation, i.e., the affine interpolation of atomic positions, leads to errors. On the other hand, the summation rule introduces energy approximation errors which, in turn, lead to residual and spurious force artifacts that relax the sample to an incorrect minimum of the (approximated) total potential energy. Together, these result in the total error of the nonlocal, energy-based, quasistatic QC method.

For the exact method of calculating elastic constant errors, see Amelang et al. [2015] and the

derivation in, e.g., [Venturini \[2010\]](#).

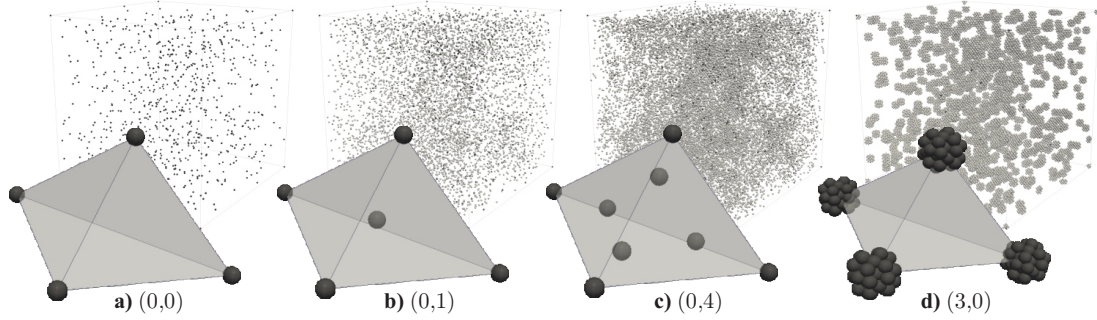


Figure 3.19: Example distributions of sampling atoms for four different summation rules applied to the same set of repatoms: a) purely nodal (0,0), b) one point quadrature (0,1), c) four point quadrature (0,4), and d) 3-shell cluster rule (3,0). Also shown is a sample tetrahedral element with its repatoms and sampling atoms for each case. The four graphics give an impression of the different levels of computational expenses which grow with the total number of sampling atoms in the QC model.

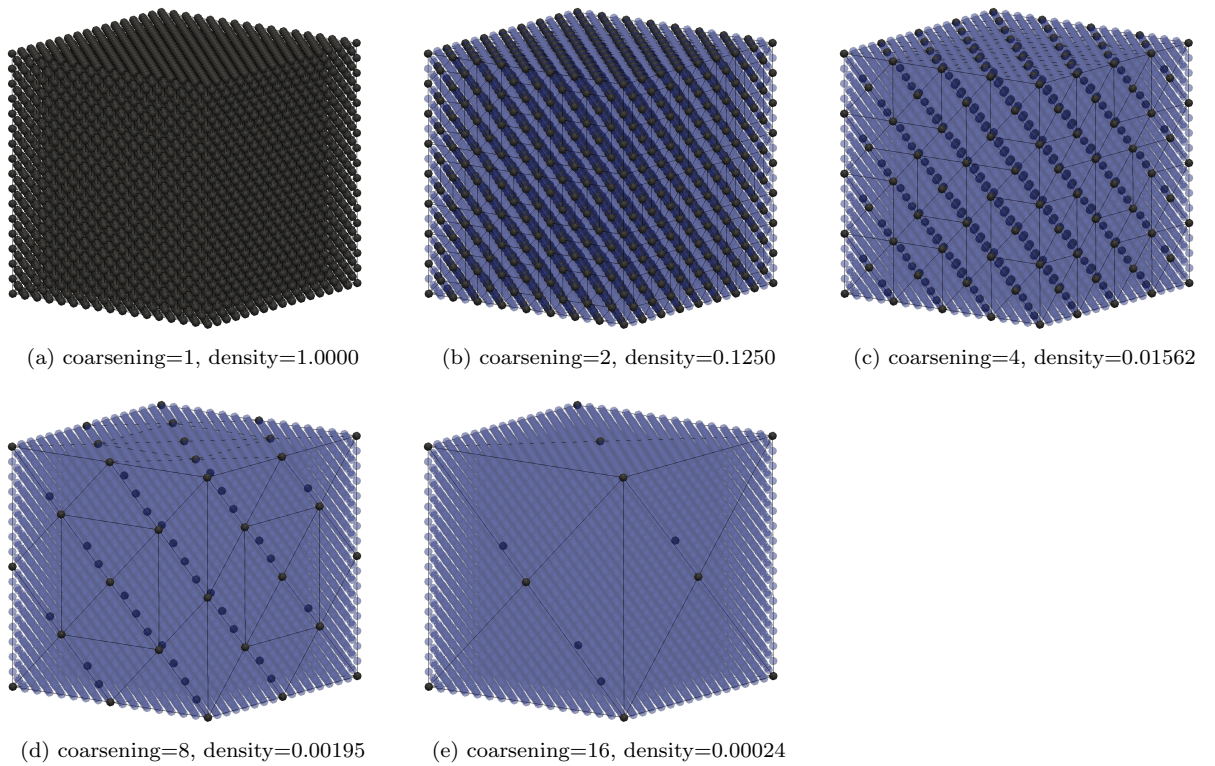


Figure 3.20: An illustration of repatom density and coarsening ratios.

Repatom densities: Before interpreting the results of this example, it is important to keep in mind the significance of the repatom density, as shown in Figure 3.20. Coarsening an atomistic cube

Figure 3.21: Results of elastic constant calculations: a)-c) relative errors of the elastic constants (computed by the QC method using various summation rules) compared to their (theoretically) exact values obtained from full atomistics; d) relative errors vs. normalized computation times required for elastic constant calculations for the different summation rules (small percentage labels denote repatom densities; computation times are normalized by those of comparable fully-atomistic run times).

Discussion of accuracy results: Figs. 3.21(a)-(c) illustrate the relative error of the three elastic constants obtained from the various summation rules vs. repatom density. A repatom density of 100% corresponds to full atomistics and is free of errors. Each point denotes the mean error of all random realizations for a specific summation rule and repatom density; the associated standard deviations are indicated by error bars. When interpreting these results, it is useful to keep in mind that repatom densities around 1-100% are extremely rare in practice, densities around 10% only occur in the immediate vicinity of full atomistic resolution, and most QC domains are of repatom density far below 1%.

Overall, the results shown in Figure 3.21 admit the following conclusions:

- Uniformly coarse-grained samples reproduce all elastic constants exactly. This can be expected since equally-spaced repatoms are free of spurious forces, which results in affine deformations and hence in a uniform atomic energy which, in turn, is summed exactly by all summation rules due to zeroth-order consistency. Consequently, energies and thus elastic constants must be reproduced exactly. Marginal errors are visible in some of the data points for uniform meshes and demonstrate the level of numerical errors to be expected from any parallel floating point calculation.
- For non-uniform repatom distributions, nodal summation rules of type $(0,0)$ exhibit the largest relative errors for all three elastic constants. For densities below 10% (not shown because of the scales in graphics), the mean error appears to stabilize in the 9 – 13% range with individual realizations showing errors well above 25%, which is generally not tolerable.
- Adding clusters to the nodal summation rule considerably reduces these errors. Cluster rule $(3,0)$ produces errors significantly below those of the nodal rule $(0,0)$. The effect is most pronounced at high density values, as can be expected: at values of 10% the average element size is very small (twice the side length of an atomistic mesh) so that clusters essentially sample at every possible lattice site. As reported by Eidel and Stukowski [2009], cluster sizes are most effective in this regime and can lead to significantly improved results. However, lower densities (for example, below 5%) result in larger average element sizes, for which the error rises rapidly. At very low densities near 0.1% the impact of clusters is marginal and the elastic constant error is comparable to that of the nodal summation rule.
- Of all traditional summation rules, quadrature rules of type $(0, n_Q)$ show the smallest relative errors for the entire range of densities investigated here. In particular, for quadrature rule $(0,1)$ the relative errors of all elastic constants remain below 5% across all investigated density levels. When using quadrature rule $(0,4)$, the relative errors remarkably remain below 1.2% throughout (for C_{11} even smaller). Interestingly, cluster summation rule $(3,0)$ outperforms

the lowest-order quadrature summation rule $(0,1)$ only for densities larger than 5%. At lower densities (where element sizes increase), $(0,1)$ provides considerably lower errors than the much more expensive cluster summation rule $(3,0)$. In the dilute limit, $(0,1)$ requires approximately $N_h + N_e$ sampling atoms (with N_e the number of elements, and we neglect the possibility of double-counting in small elements). The number of elements in simplicial 3D meshes scales roughly linearly with the number of nodes. In those meshes shown here we observe $N_e \approx 5 N_h$, which results in an approximate total number of sampling atoms of $6 N_h$. In contrast, $(3,0)$ applied to, for example, an fcc Cu lattice requires approximately $42 N_h$ sampling atoms (because clusters do not overlap in the dilute case).

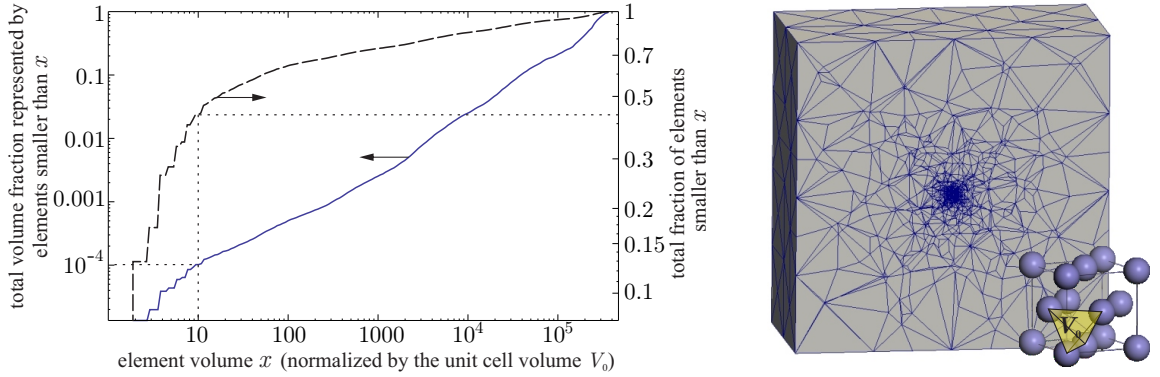


Figure 3.22: Distribution of element sizes in the void-expansion simulation shown on the right: without considering all fully-atomistic elements, the curves indicate what volume fraction of the entire domain is occupied by elements smaller than a particular size (normalized by the smallest (atomistic) element size, $V_0 = a_0^3/24$, shown schematically for fcc), and the fraction of the number of elements smaller than a particular size.

- The new first-order summation rule $(0,1^*)$ shows noticeable errors only at reparam densities above 10%, and even these remain below 1.4% for all elastic constants. Remarkably, below densities of 10% the error becomes marginal and is comparable to the numerical errors found in uniform QC meshes. This holds true for all three elastic constants. The largest errors appear for reparam densities close to 100%, which, in an actual QC simulation, primarily affects elements in the transition region right above fully-atomistic resolution. In our general QC framework with mesh adaptation the simulation volume occupied by elements of such sizes is rather small. This is illustrated in Figure 3.22 for the example of a Cu single crystal with a nano-void at its center with fully-atomistic resolution around the void and diffuse coarse-graining away from the void. Here, it is difficult to talk about an individual element's reparam density, which is why we plot the curves vs. the element volume normalized by the atomic cell size V_0 (i.e., a normalized element size of 10 implies a reparam density of $1/10 = 10\%$ if the entire domain was filled with such elements). As shown by the dotted lines, in this example about 43% of all elements have reparam densities larger than 10% (and can thus be expected

to show errors in their elastic behavior on the order of 1%). Yet, due to their small volume, these elements occupy about 0.01% of the overall simulation domain (which contains a total of 42568 elements).

The new second-order rule $(0, 1^*, 4^*)$ shows even lower errors in general simulations. However, the random meshes used for the elastic constant simulations contain extremely-poor tetrahedral elements for which reasonable definitions of the weights of face-centered sampling atoms become nontrivial. This problem does not arise in reasonable meshes as those obtained, for example, from our adaptive remeshing (see Section 3.4).

Discussion of cost results: To assess the computational cost associated with each summation rule, Figure 3.21(d) shows elastic constant errors vs. computational runtime per lattice site. Specifically, we compare the computation time required to obtain the energy curves from which C_{44} is extracted (which involves applying an average strain $\epsilon = \gamma \xi$ to the sample and relaxing the system to obtain its energy for $\gamma_{\min} < \gamma < \gamma_{\max}$). We note that in our implementation there is no algorithmic difference between the various summation rules during these steps. Once the neighborhoods and weights of each sampling atom are generated, solver iterations treat all sampling atoms the same. The only difference is found in the initial setup of the QC representation, which is not considered here.

The difference in runtime for each summation rule is a function of two factors: the complexity of the summation rule and how many solver iterations were taken. The results shown in Figure 3.21d lump both factors together. Figure 3.21(d) contains the same relative errors shown in Figure 3.21(b) but plotted vs. the computational run time normalized by the computation time of a comparable fully-atomistic calculation using the same code to allow for a fair comparison (this accounts for the increasing domain size with decreasing density). The small percentage values indicate the repatom densities for reference. As can be expected, for each summation rule the computation time per lattice site generally decreases with decreasing density. This is a direct consequence of the repatom count reduction, which automatically leads to a reduction of the number of sampling atoms in each scheme. Interestingly, at high repatom densities, the new summation rule $(0, 1^*)$ – just like the cluster rule $(3, 0)$ – is computationally more expensive than full atomistics, primarily because of the large number of sampling atoms. At lower densities, however, the graphic shows that the new rule provides the lowest errors and low computational run times.

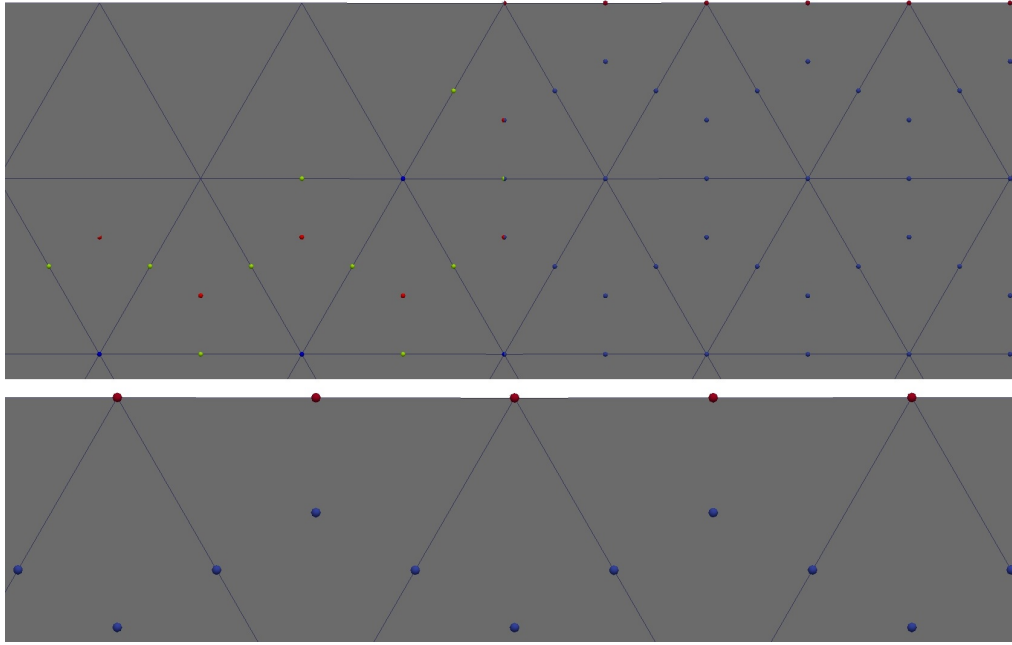
Overall, the quadrature summation rules $(0, 1)$ and $(0, 4)$ perform much better than the nodal and cluster summation rule $(3, 0)$, showing lower errors at comparable or lower costs. If the relative error is required to remain below 3%, $(0, 1)$ is sufficient and by far the most efficient choice. For the relative error to remain below 1%, $(0, 4)$ is required and, again, by far the most efficient summation rule. As seen before, cluster rule $(3, 0)$ only reduces the error in case of high repatom densities and

otherwise produces large relative errors. $(3, 0)$ is considerably (by orders of magnitude) less efficient than the pure nodal rule $(0, 0)$. The seemingly interesting behavior for $(3, 0)$ at large runtimes stems from overlapping clusters becoming significant at large levels of the repatom density. The new scheme $(0, 1^*)$ incurs costs that are comparable to the $(0, 1)$ quadrature rule at lower densities. It is more expensive at high repatom densities because inner-element sampling atoms persist until their weights go to 0, whereas quadrature rules (which snap sampling atoms to actual lattice sites) produce fewer sampling atoms near the atomistic limit. At low repatom densities, the errors of $(0, 1^*)$ are so small that they can no longer be shown on the logarithmic scale while becoming less and less costly. Therefore, at low repatom densities of 1% and below, the by far most efficient and most accurate choice in our examples is the new $(0, 1^*)$ scheme.

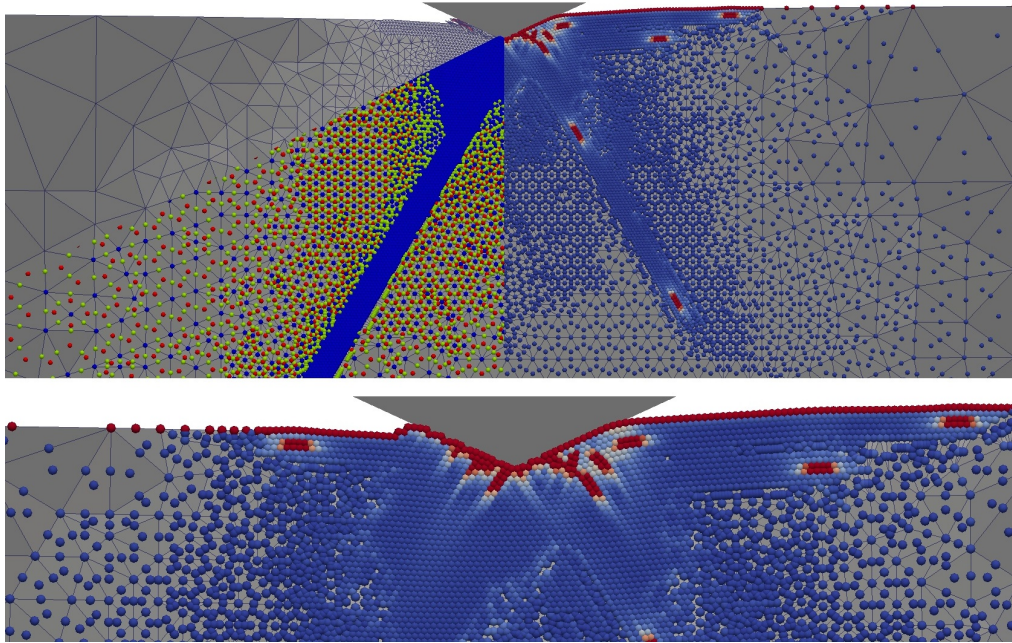
3.4 Inelasticity and the formation of microstructure

Nanoindentation is a classical example that has been subject to previous investigations by the QC method, see for example [Smith et al., 2001, Knap and Ortiz, 2003, Jin et al., 2008]. Most previous studies considered relatively small sample and indenter sizes or incipient indentation depth. Here, we apply the presented fully-nonlocal QC formulation with the new second-order summation rule to two- and three-dimensional single-crystalline Cu samples deforming under spherical indenters modeled by the potential of Kelchner et al. [1998]. This scenario goes beyond our previous examples in various ways. On the one hand, indentation leads to significant inelastic deformation mechanisms involving dislocation nucleation and propagation. Here, the nature of the atomistic energy landscape with myriads of local minima corrupts the repeatability of results in the presence of numerical noise (due to computational parallelism). Therefore, we qualitatively compare the microstructure obtained from QC to that of full lattice statics; quantitatively we compare the load vs. indentation depth curves via statistical averaging as explained below.

Also, this scenario involves free surfaces which present a challenge to the QC method because of considerable force artifacts arising from the traditional summation rules. Using, for example, a cluster summation rule results in large residual forces which lead to an artificially deforming free surface. This in turn will initiate mesh adaptation and ultimately lead to atomistic refinement of the entire surface, which is prohibitively expensive. In contrast, the new second-order rule assigns sampling atoms to the faces of tetrahedral elements, which automatically takes care of free surfaces by representing all surface-nearest lattice sites by those sampling atoms on the surface. We will report details of how surfaces and interfaces can be modeled by the new scheme in the examples of Section 3.5; here we will show that the second-order scheme can successfully master the combined challenges of free surfaces and microstructure formation.



(a) Initial condition



(b) Deformation at peak indentation depth (3nm)

Figure 3.23: 2D quasistatic nanoindentation example performed with automatic mesh refinement. The mesh is colored by the element size. Sampling atoms are colored by either centrosymmetry (blue-red spectrum) to highlight dislocations or by their sampling atom type (blue, green, or red). A movie of the simulation is available [here](#) for a quasistatic solution and [here](#) for a solution using implicit dynamics.

3.4.1 2D nanoindentation

While a two-dimensional setting allows for a direct comparison with full atomistics, the three-dimensional example by far exceeds our computational resources to run a comparative lattice statics simulation. Therefore, we will first illustrate results obtained in two dimensions. A Cu sample (i.e., a close-packed $\{111\}$ plane modeled by the extended Finnis-Sinclair potential of Dai et al. [2006]) of dimensions 0.1×0.082 microns and containing 148,000 lattice sites is relaxed and then indented by a circular indenter with a 20 nm diameter and a force constant of 10^3 eV/Å³. The sample's bottom is constrained to suppress displacements in the vertical direction, whereas lateral edges are constrained in the horizontal direction. The quasistatic evolution of the indenter is simulated by 1,000 displacement steps at increments of 0.003 nm (i.e., each relaxation step involves about 1% of the Cu lattice spacing). Movies of a similar scenario are available [here](#) and [here](#), for context.

Twenty realizations have been used to obtain the average force-indentation curves shown in Figure 3.24. The parallel implementation of the QC method (specifically, the random order in which threads contribute to the repatom forces) leads to small random fluctuations, which in turn produce variations of the simulated results (similar to the effect of thermal fluctuations). Therefore, we show the average curve along with its standard deviation.

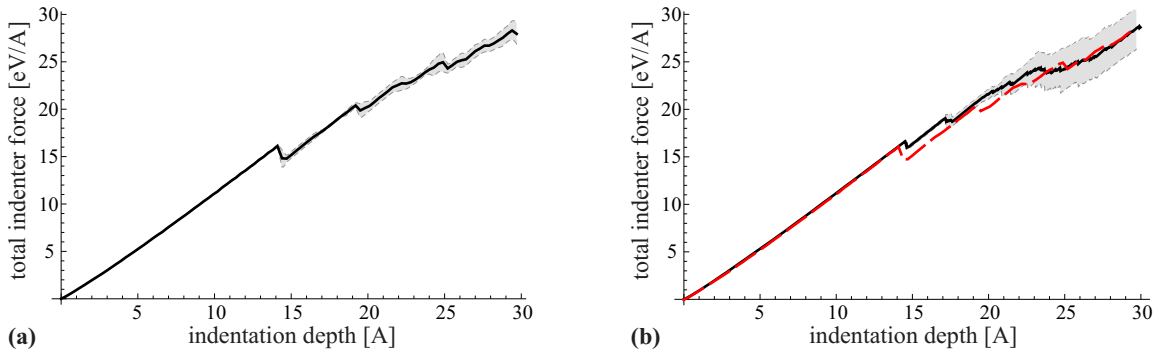


Figure 3.24: Indenter force vs. indentation depth for a two-dimensional Cu single crystal obtained from (a) lattice statics (full atomistic resolution) and (b) the QC method using the new second-order summation rule and automatic mesh adaptation. Shown are the mean force-displacement curves obtained from 20 simulations with standard deviations shown by the gray shading. For a direct comparison, the dashed (red) line in (b) is the average response from (a) obtained from lattice statics.

Overall, the QC results agree well with the lattice statics findings. In particular, the initial elastic response is identical, the first dislocation event occurs at identical indentation depths, and the general trends of the load-displacement curves show great similarities. It is worth mentioning that our QC simulation does not use a fixed mesh with a large atomistic region underneath the indenter, but instead we use automatic mesh adaptation to locally refine the initial QC mesh. To this end, we use the second-invariant criterion of [Knap and Ortiz, 2001] with a tolerance of 0.1 to identify regions of refinement (i.e., we check if the second invariant of an element's Lagrangian

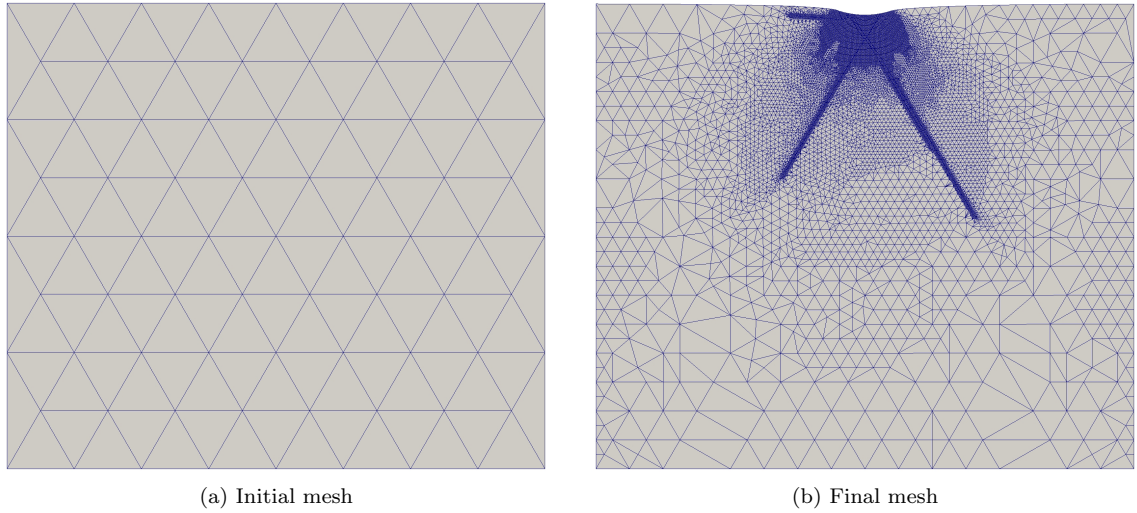


Figure 3.25: Initial and final QC meshes of a nanoindentation simulation in two dimensions using automatic mesh adaptation. Movies are available [here](#) and [here](#).

strain tensor exceeds its threshold). We then apply a complex combination of element bisection and local mesh reconstruction for mesh refinement, a strategy that will be discussed in the section on model adaptation (4.8). Reduction of the tolerance parameter below 0.1 increased atomistic refinement but appeared to produce no significant differences in the average force–displacement curves. Figure 3.25 shows the initial QC mesh as well as the final QC representation at an indentation depth of approximately 3 nm.

In order to better compare the forming microstructure, Figure 3.26 shows the final deformed state at an indentation depth of about 3 nm, color-coded by the centrosymmetry parameter of individual atoms [Kelchner et al., 1998]. For the QC solution we first determine the positions of all lattice sites from those of the repatoms via the affine QC interpolation. Then, we compute the exact centrosymmetry parameter for each atom in the interpolated configuration. Interpolations may lead to non-centrosymmetric neighborhoods across elements, which is why Figure 3.26b shows significantly more noise than the exact solution in Figure 3.26a. It is important to note that this noise does not represent actual defects but simply interpolation errors that would disappear if the final QC positions were relaxed without the affine interpolation constraints. Therefore, we are primarily concerned with the qualitative agreement of the observed dislocation activity. We see that all three slip systems (in the close-packed directions of the 2D lattice) and the exact same slip planes are activated in both solutions; the pile-up of dislocations directly underneath the indenter agrees qualitatively, and the core structure and size of dislocations is identical in both simulations. Of course, we do not claim a perfect fit but the qualitative agreement is convincing in our view, considering that this simulation uses automatic mesh adaptation and started from the mesh of

Figure 3.25a. Also, unlike in many other atomistic-to-continuum coupling techniques there is no methodological differentiation between atomistic and coarse-grained regions.

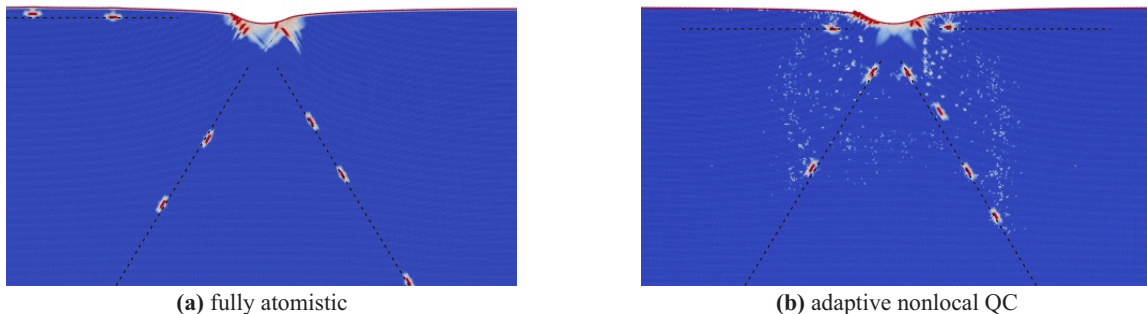


Figure 3.26: Dislocation emission from the surface during nanoindentation simulated by (a) lattice statics and (b) the fully-nonlinear energy-based QC method with the new second-order summation rule and automatic mesh adaptation in two dimensions (see the QC meshes in Figure 3.25), color-coded by centrosymmetry on a log-scale (0.07-0.7, red being highest). Noise in the QC solution stems from the affine interpolation scheme.

In addition spherical indenters, we have explored the microstructure formed by pyramidal indenters. A 2D example with an indenter that forms an angle of 1.14 radians from the vertical axis is shown in Figure 3.23 and can be seen as a movie [here](#). Figure 3.27 shows force-displacement curves for both spherical and pyramidal indenters. As can be expected, indentation force increases with increasing cross-sectional area for a given indentation depth. The differences in curves for the same geometry stem from noise introduced by parallel floating point calculations.

The above 2D simulations are performed quasistatically and every frame is a different loadstep (a complete equilibration), which is why dislocations move great distances between movie frames. By using implicit dynamics, the distance that dislocations move can be limited in order to study the formation of microstructure instead of fully-equilibrated configurations. A movie showing the behavior of remeshing, adaptive neighborhoods, and implicit dynamics can be seen [here](#).

3.4.2 Large-scale 3D nanoindentation

Finally, let us demonstrate the capabilities of the presented energy-based QC flavor by a three-dimensional example. Figures 3.28 and 3.29 display the evolution of dislocation loops underneath a 40 nm indenter in three dimensions, using the indenter potential of [Kelchner et al. \[1998\]](#) with a force constant of 10^3 eV/A³. The maximum penetration depth is 6 nm, and the sample size is 1 micron (containing approximately 86 billion lattice sites which are represented in the final deformed state by an unprecedented 16.8 million degrees of freedom). Due to the large sample size, we do not perform comparative simulations using full lattice statics and instead only report our findings from the QC simulation. To avoid the distraction by approximation errors stemming from the remeshing scheme (and to confirm that the scheme can equally well handle sharp mesh transitions), we start with a

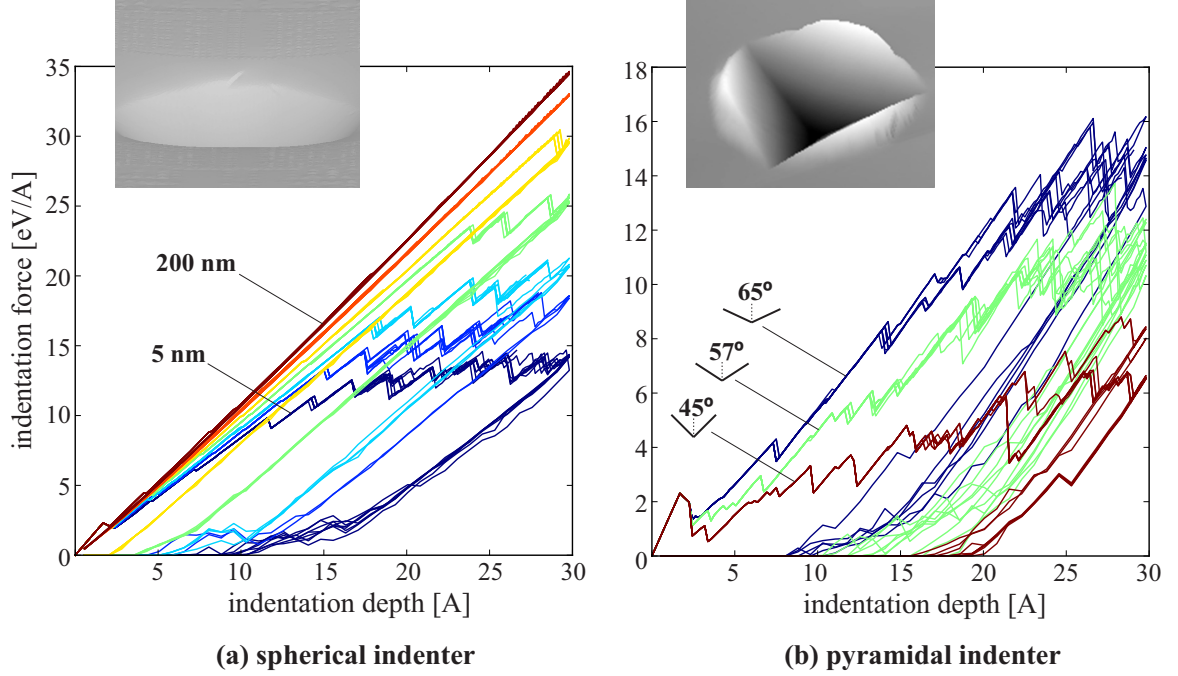


Figure 3.27: Force-displacement curves of many nanoindentation realizations for spherical (left) and pyramidal (right) indenters.

large atomistic region, as shown in Figure 3.28.

The Cu micro-cube is initially single-crystalline and defect-free but contains a spherical void of radius 0.75 nm at a depth of 4 nm below the indented surface; such voids arise, e.g., from exposure to radiation [Whitley et al., 1979]. The void creates an inhomogeneous stress field with concentrations in its vicinity, which is of interest for two reasons. First, this promotes the heterogeneous nucleation and emission of dislocation loops from the void (the large indenter size would require considerably larger indentation depths to create defects in an initially defect-free crystal). Second, the interaction of the void with the free surface is an ideal scenario for the QC method since it requires to accurately capture long-range elastic fields. Void growth in infinite crystals subject to remote loading has been investigated by the QC method before, see for example [Marian et al., 2005, 2008, Ariza et al., 2012], but not in a macroscopic boundary value problem such as the nanoindentation scenario studied here. Results of the microstructure evolution from the initial void to the final defect network are shown in Figure 3.29 (defects have been identified by the centrosymmetry parameter).

3.5 Free Surfaces

As described in Section 1.4, surface effects in nanoscale mechanical systems are of great interest to the scientific community and present a major challenge to coarse-grained atomistics. The fully-nonlocal energy-based QC method described in this thesis is well-qualified, through its new second-order

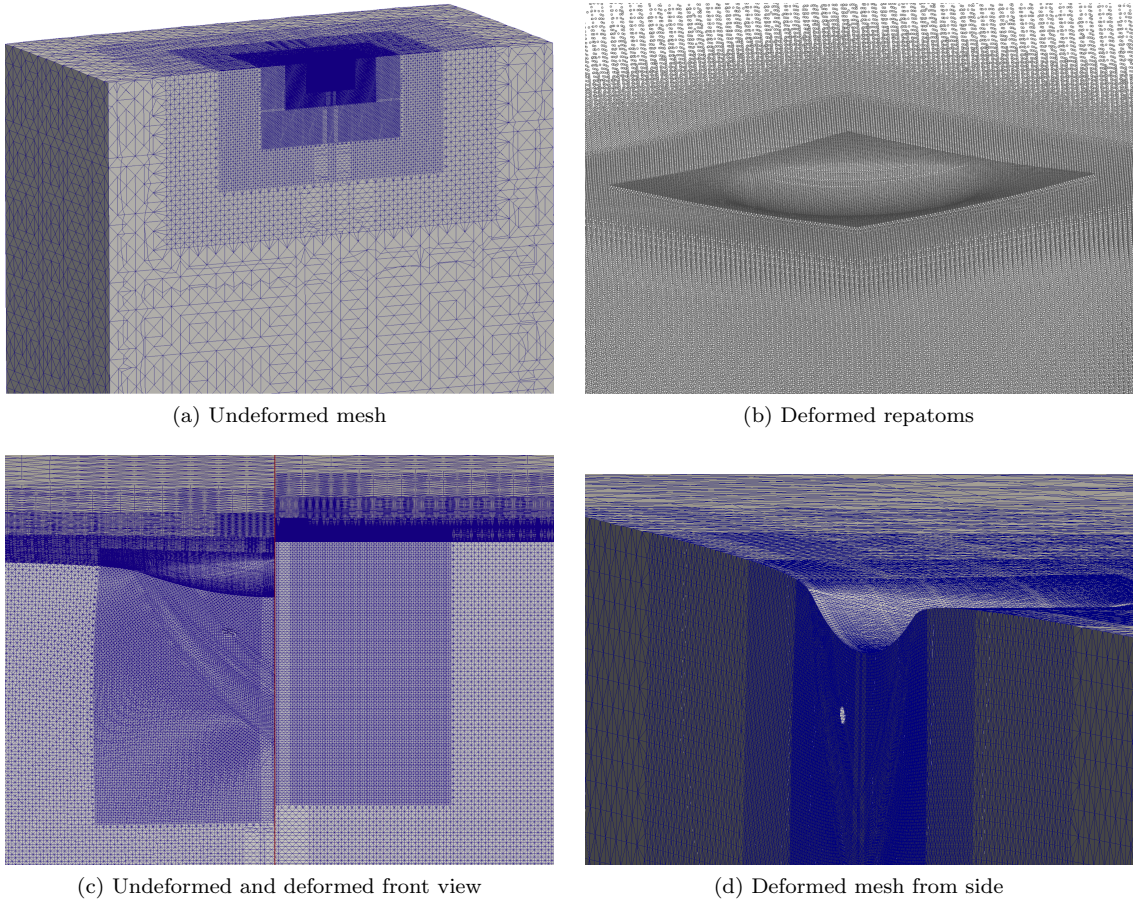


Figure 3.28: Indentation of a spherical indenter into a nano-cube of 86 billion lattice sites, represented by 16.8 million quasicontinuum degrees of freedom.

optimal summation rule, to describe free surfaces. In this section, we show that our new coarse-graining scheme successfully and automatically reduces spurious force artifacts near free surfaces. We demonstrate its benefits in the presence of free surfaces by several nanomechanical examples including surface energy calculations, elastic size effects in nano-rods, and in plates with nano-sized holes.

The relaxation of free surfaces leads to a local rearrangement of atomic positions in the crystal lattice due to a loss of symmetry in atomic neighborhoods near the surface, see for example [Rous, 1995, Inglesfield, 1995] for comprehensive overviews. Depending on the atomic structure and interactions, surface relaxation can be as simple as changing only the spacing between atomic layers near the surface, but it can also involve a complex surface reconstruction (oftentimes seen in transition metals). Here, we restrict our attention to non-reconstructing surfaces of metals which result in a change of the spacing between atomic layers close to the free surface; see Figure 3.30a. That is, atomic positions are relaxed normal to the surface.

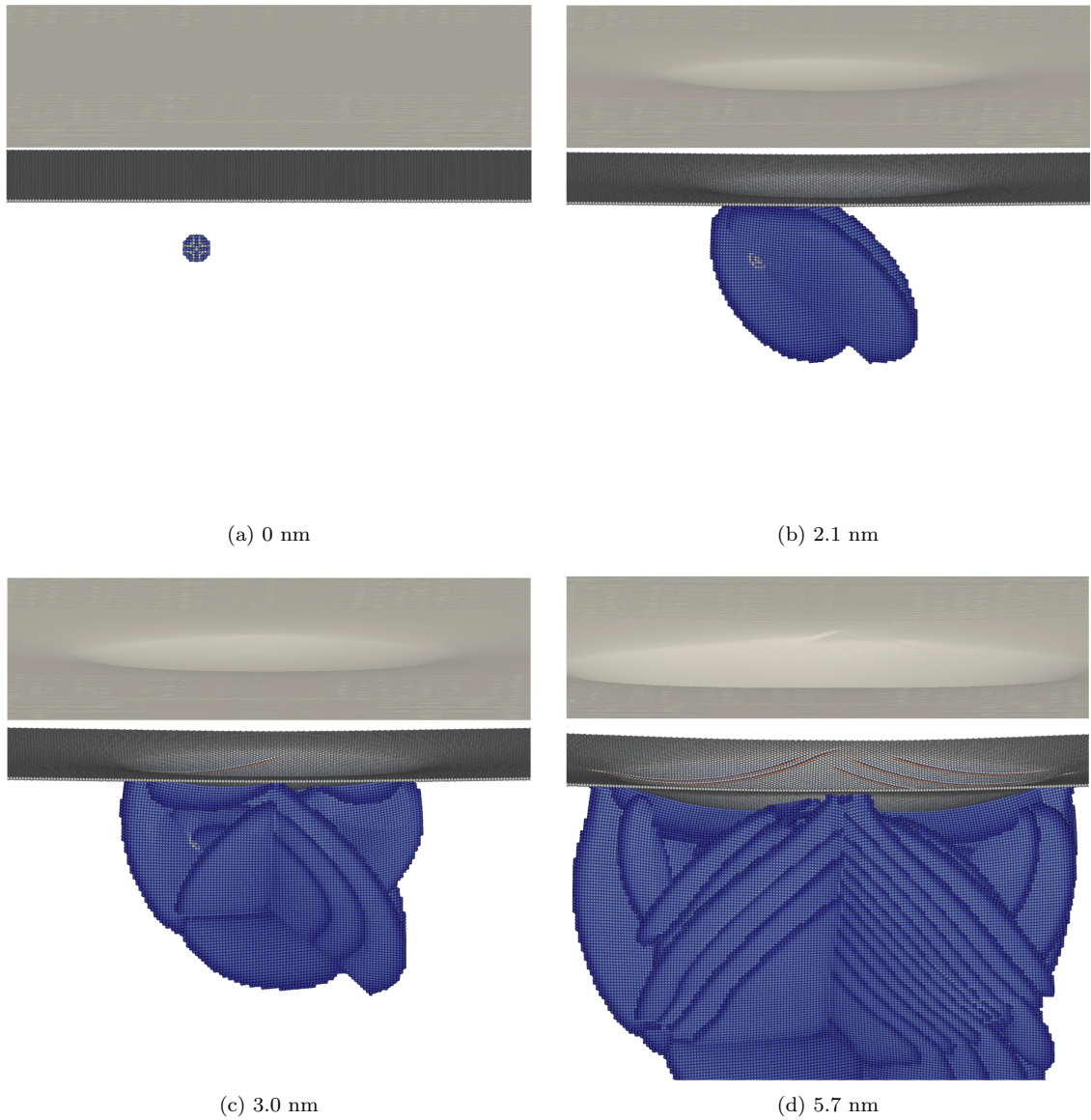
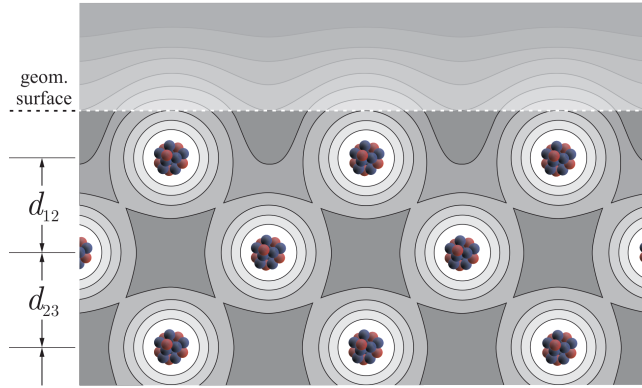
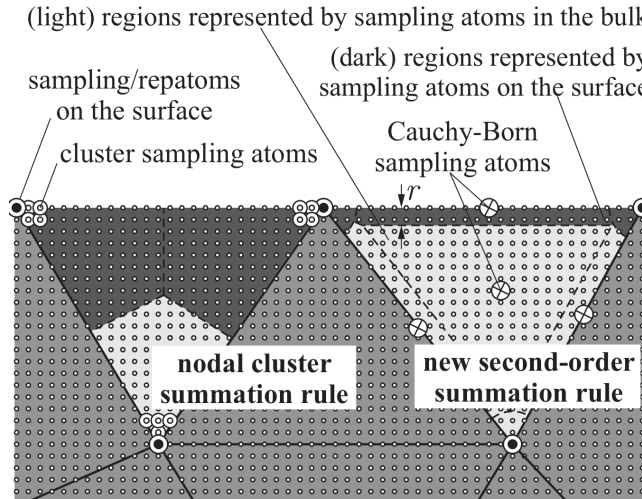


Figure 3.29: Zoomed-in snapshots of the microstructure evolving during 3D nanoindentation into a Cu single-crystal with an isolated nano-void at indentation depths of 0 nm, 2.1 nm, 3.0 nm, and 5.7 nm. The top half of each panel shows a side view of the deformed mesh, the lower half shows the atomistic repatoms with high centrosymmetry (color code in arbitrary units).

For most metals, this phenomenon affects primarily the distance between the first few atomic layers, although it can extend deeper depending on the material. Also, most metals exhibit a contraction of the atomic layers; only group 2 and 3 metals (for example, beryllium and aluminum) have been shown to produce an expansion of the surface layer spacing. To what extent atomic layers are affected by the surface relaxation depends on the effective range of atomic interactions (in the following denoted by r), which in simulations are related to the cut-off radius of the potential (the



(a) Schematic view of the structure of a free surface in a metallic crystal: positions of atomic cores and the corresponding electron density distribution with the characteristic spacings of atomic layers below the geometric surface (d_{ij} deviate from the lattice spacing a_0 near the surface).



(b) Visualization of a classical nodal cluster summation rule and the new second-order summation rule along with their repatoms and sampling atoms in simplicial elements near the surface (dark regions represent those lattice sites represented by sampling atoms on or near the surface for the example of an effective atomic interaction distance of $r = 1.5a_0$).

Figure 3.30: Schematic views of metallic free surfaces and pertinent QC summation rules.

effective interaction distances may be shorter due to long tails of the potentials).

In the following, we tacitly assume that the chosen interatomic potential can capture surface effects with sufficient accuracy. In this case, a full atomistic simulation can easily determine the relaxed surface configuration by equilibrating the atomistic ensemble in the presence of free surfaces. Depending on the material and the surface orientation, this geometric relaxation effect might be rather small; for example, changes in atomic layer spacings below planar surfaces in pure Cu range between 1 – 10% for the first layer (for different surface orientations) and decay quickly with

increasing depth [Rous, 1995]. However, the presence of a surface significantly affects the mechanics of those layers nearest to the surface. Leaving aside the various inelastic surface-dominated mechanisms (for example, the attraction of moving dislocations, single-arm dislocation sources, or crack initiation), the elastic performance of the crystal near the surface gives rise to size effects in the elastic moduli.

When using the QC method to model small-scale structures, the coarse-graining of crystalline regions near free surfaces creates problems. Of course, the chosen interpolation within elements prevents individual atomic layers from adjusting their equilibrium spacing near the surface (but, as argued above, this geometric effect is rather small). However, a considerable error arises from the introduction of summation rules if all inner-element lattice sites are represented by sampling atoms on or near the surface (such as in nodal or node-based cluster summation rules), or if all surface-atoms are represented by inner-element sampling sites whose energies remain unaffected by the surface (such as in the classical local QC formulation using the Cauchy-Born rule for all coarse-grained elements). Figure 3.30b visualizes the distribution of sampling atoms for both a traditional nodal cluster summation rule with tessellated weights and the new second-order summation rule within elements adjacent to the free surface. The new scheme achieves a highly-accurate representation of surface-nearest lattice sites by sampling atoms on the surface. Alternatively, many previous QC simulations have had to make use of full atomistic refinement near the surface, see [Fang and Yang, 2014] for a recent example. While this approach is most accurate, it also incurs tremendous computational expenses that become prohibitive when modeling large, above-nanoscale systems.

The new second-order summation rule avoids this problem through sampling atoms on all element faces (or edges in 2D). Their weights are chosen such that they automatically represent only those face-nearest lattice sites whose energies deviate from that in the bulk. Of course, this accounts for free surfaces only in an average sense (all surface-nearest atoms are approximated by the same energy per atom). Yet, the scheme ensures that (i) the total potential energy is approximated fairly well by accounting for both surface and bulk atoms, and (ii) only minimal force artifacts arise from free surface in non-uniform QC meshes. As we will show in the following examples, the latter is an important observation because, for example, a purely nodal summation rule can lead to large spurious deformation of the free surfaces in response to residual force artifacts. Also, residual forces can no longer be identified as such since the undeformed configuration of a non-uniform QC mesh now exhibits both non-physical force artifacts and real relaxational forces near the surface, the distinction between which is impossible without running a full MD simulation for comparison. Using the incorrect ad-hoc method of precomputing residual forces and removing them as dead loads from repatoms in arbitrary deformed configurations would eliminate the effect of the free surface entirely.

The above discussions have shown that the fully-nonlocal energy-based QC methodology along with the new second-order summation rule presented in this thesis seems well suited to capture

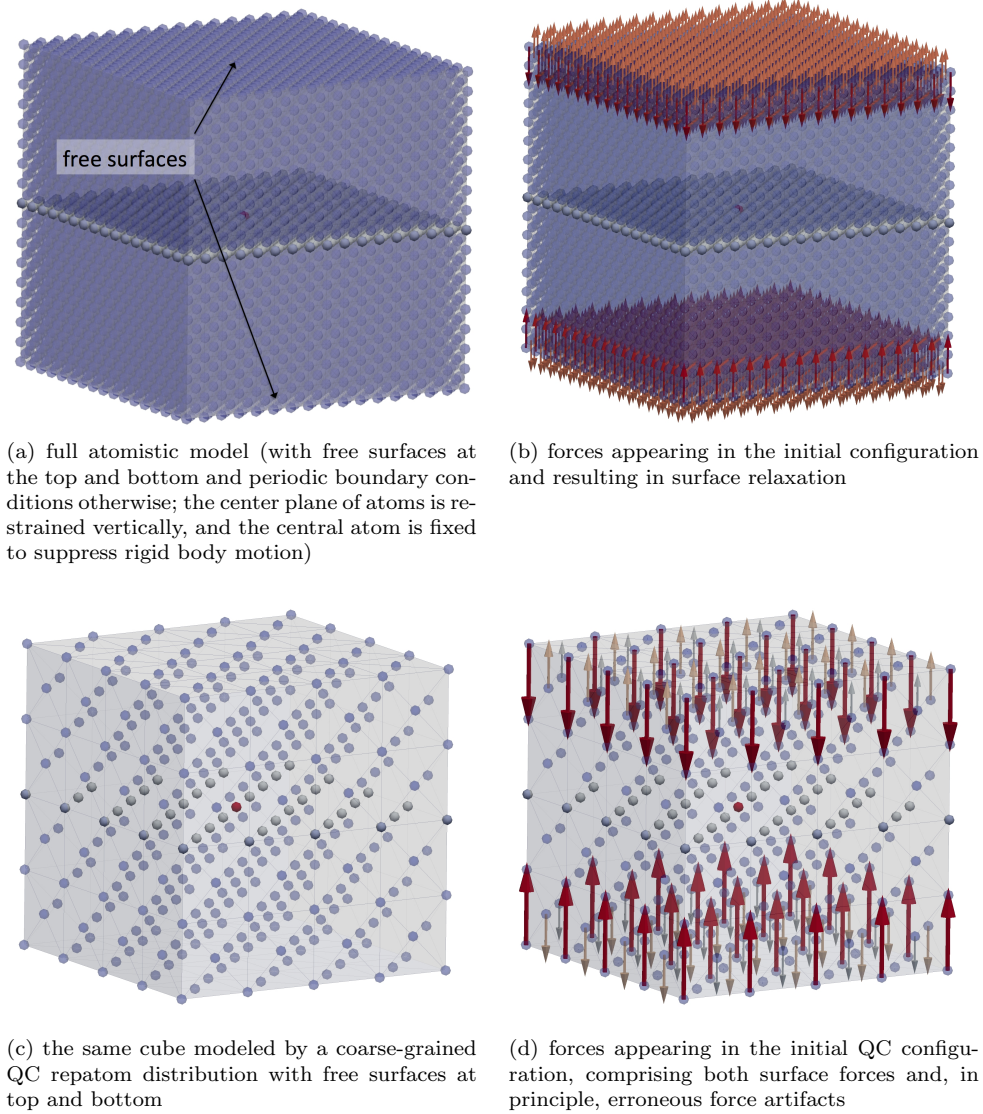


Figure 3.31: Schematic view of the model used to determine surface energies.

surface effects in nanoscale structures. In the following, we report a series of computational studies that probe elastic size effects and inelastic surface effects in small-scale structures.

3.5.1 Calculation of surface energies

The calculation of surface energies by molecular statics (or dynamics) is straight-forward and can provide reliable data when suitable interatomic potentials are being used; see for example [Todd and Lynden-Bell, 1993] for a representative study. In contrast, coarse-grained atomistic simulations commonly fail to accurately capture such surface characteristics due to (i) spurious force artifacts near the surface which lead to an incorrect deformation of the atomic layers near the surface, and

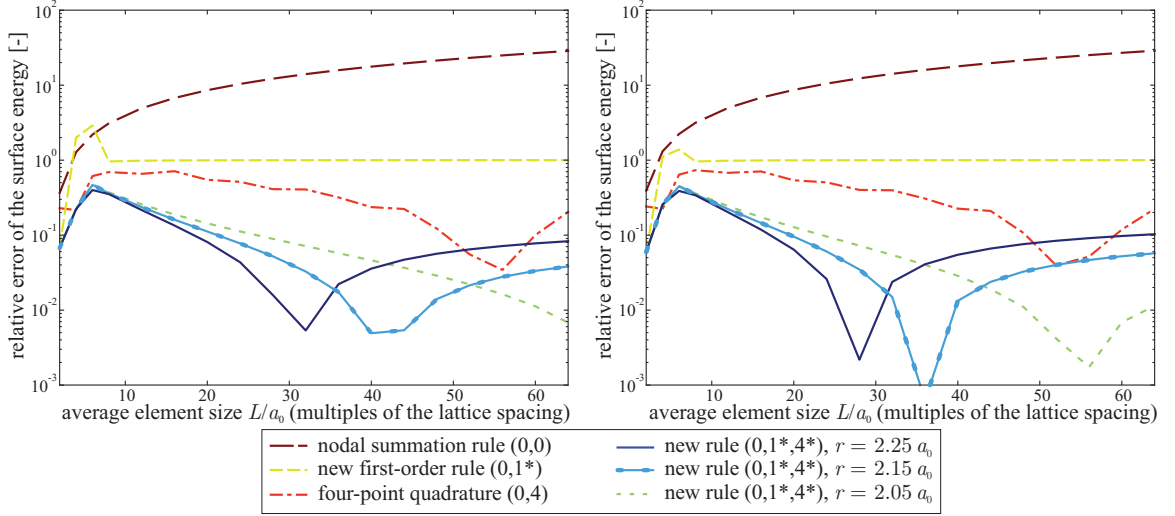


Figure 3.32: Relative error of the computed surface energy vs. QC element size for single-crystalline (100) Cu modeled by the potentials of (a) Mishin et al. [2001] and (b) Dai et al. [2006] for various summation rules, including the new second-order rule with different effective interaction radii r .

(ii) the errors made by approximating the potential energy through summation rules. Consequently, the total energy of QC samples with free surfaces may contain significant errors. Therefore, surface energies are a good indicator for the accuracy of our coarse-grained atomistic description.

As an example, we compute the surface energy of (100) pure single-crystalline Cu modeled by the potentials of extended Finnis-Sinclair (Dai et al. [2006]) and Mishin (Mishin et al. [2001]) at zero temperature. To this end, we construct a square slab with planar top and bottom (100) surfaces and with periodic boundary conditions imposed on the lateral faces, see Figure 3.31. We equilibrate the nano-crystal to identify all relaxed atomic positions (i) by molecular statics and (ii) by the QC method with the various summation rules. We compute the energy of the relaxed sample and contrast it with the energy of the same number of atoms in an infinite single crystal; the excess energy divided by the total surface area provides the surface energy. In addition to the choice of the summation rule, results generally also depend on the mesh size and quality.

Figure 3.32 shows the relative error of the computed surface energy as a function of the average element size in the QC representation for six different summation rules, including the new schemes of first and second order. The new rule of second order shows considerably lower errors than the other schemes across all element sizes tested, especially in the limit of large elements where the error is as low as a few percent (while comparable other schemes exhibit errors on the order of 100% and above). Of course, the error depends essentially on the choice of the effective interaction radius of the chosen potential (which does not have to agree with the potential cut-off due to long shallow tails of the most common potentials). Therefore, Figure 3.33 illustrates the error as a function of the element size and of the effective interaction radius r , showing optimal r -values near $r = 2a_0$ in

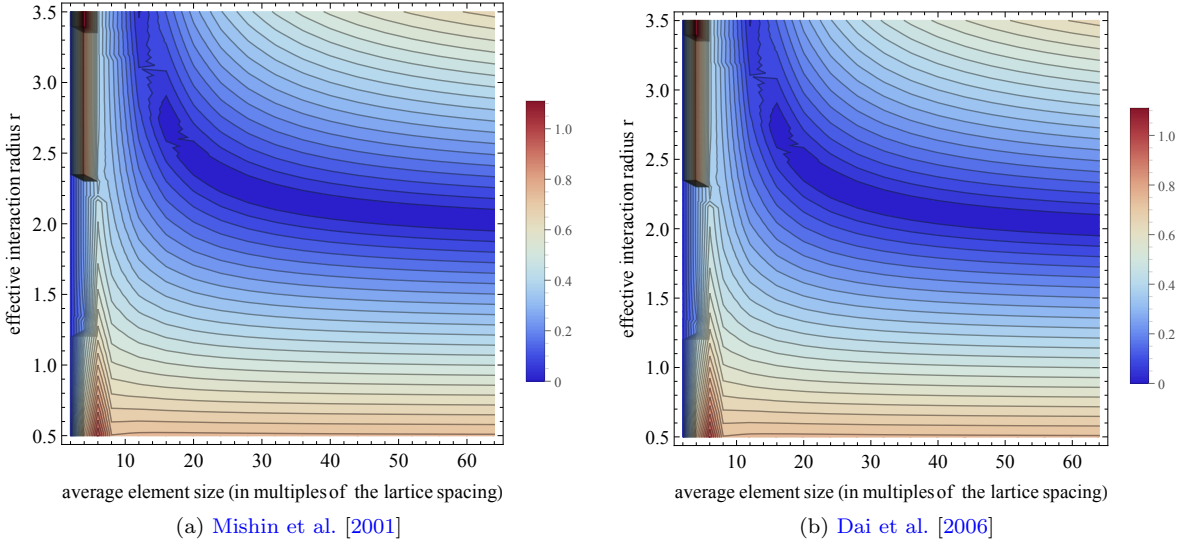


Figure 3.33: Contour plots of the relative error of the surface energy as a function of the effective interaction radius r and the average element size L/a_0 (in multiples of the atomic spacing), showing convergence for large elements.

the limit of large elements with a major deviation only apparent for element sizes between $10 - 20a_0$. This allows us to define the optimal values for the effective interaction radius r .

We note that the QC scheme has been tested with the same other potentials as in Section 3.3.1, but the results are not shown here for brevity because the same trends have been observed. It is important to keep in mind that the potential cut-off radius enters our QC model in two very distinct ways, only one of which is our focus here. On the one hand, the cut-off radius is essential for the calculation of interatomic forces as in every atomistic model. This actual cut-off radius comes with a potential and is not modified here (changing this cut-off radius may lead to significant changes in material properties). On the other hand, our new first- and second-order summation rules define the weights of sampling atoms in dependence of an *effective* cutoff radius (which is a good measure for how many atoms in the vicinity of, for example, an element node are best represented by the lattice site located at that node rather than those lattice sites within elements). This is different from all previous summation rules which defined sampling/repatom weights based on consistency or based on geometric arguments. Here, the sampling atom weights depend on the effective interaction radius r , and the consequences of changing r for weight calculations (not for interatomic force calculations) are shown in Fig. 3.33.

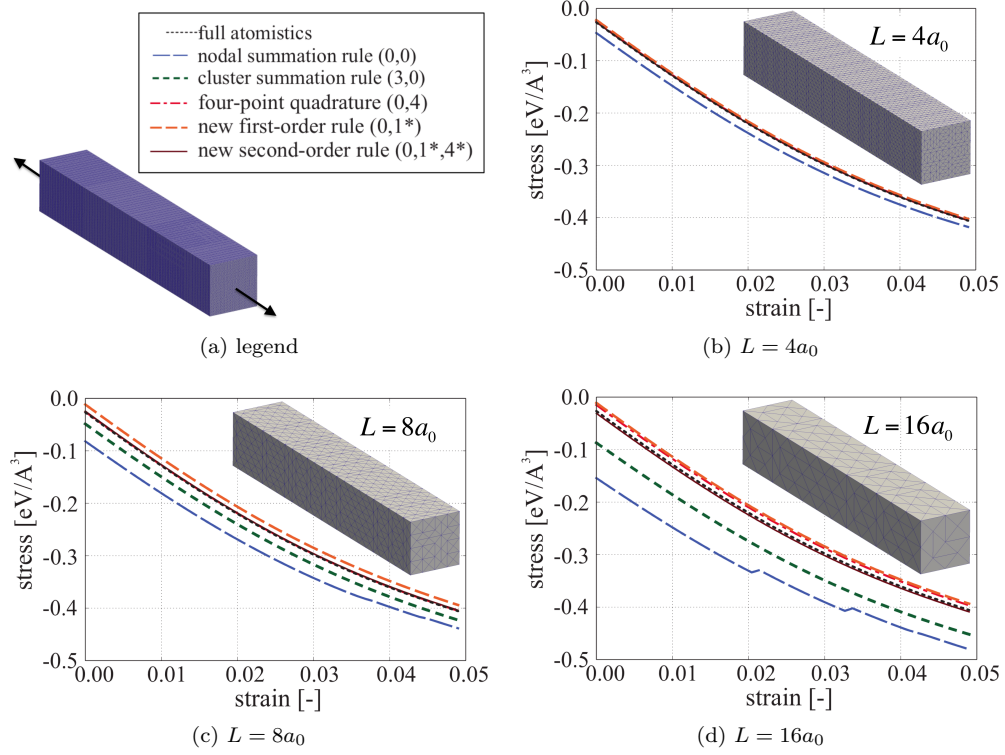


Figure 3.34: Stress-strain response of nano-rods of different sizes modeled by the QC method with five different summation rules and three different average element sizes L . The rod is stretched uniaxially with unconstrained lateral surfaces; the full atomistic solution is included for comparison. (Note that end effects are responsible for the observed initial stress at zero applied average strain.)

3.5.2 Elastic modulus of a nano-rod

In order to assess the impact of free surfaces on the elastic response of nanoscale structures, we simulate the uniaxial extension of a nano-rod with square cross-section (while circular cross-sections are more realistic and common in nano-wire experiments, the flat surfaces admit a cleaner investigation without complicating geometric effects). We deliberately consider only elastic deformation here to avoid the formation of microstructure which quickly leads to a randomized response, so that results cannot be compared directly to full atomistics.

For various summation rules and mesh sizes, we compute the load–displacement curve for the uniaxial extension of differently-sized whiskers whose length is six times the cross-sectional side length (the latter is varied to demonstrate size effects). From the resulting stress–strain relations, we extract the effective, size-dependent (structural) Young’s modulus (linearized about the undeformed state) to quantify the impact of free surfaces on the elastic response, cf. [Miller and Shenoy, 2000].

Figure 3.34 illustrates the stress-strain response of the rod for six different summation rules and for three different mesh sizes of the QC representation (a side length of $L = 4a_0$ implies that element

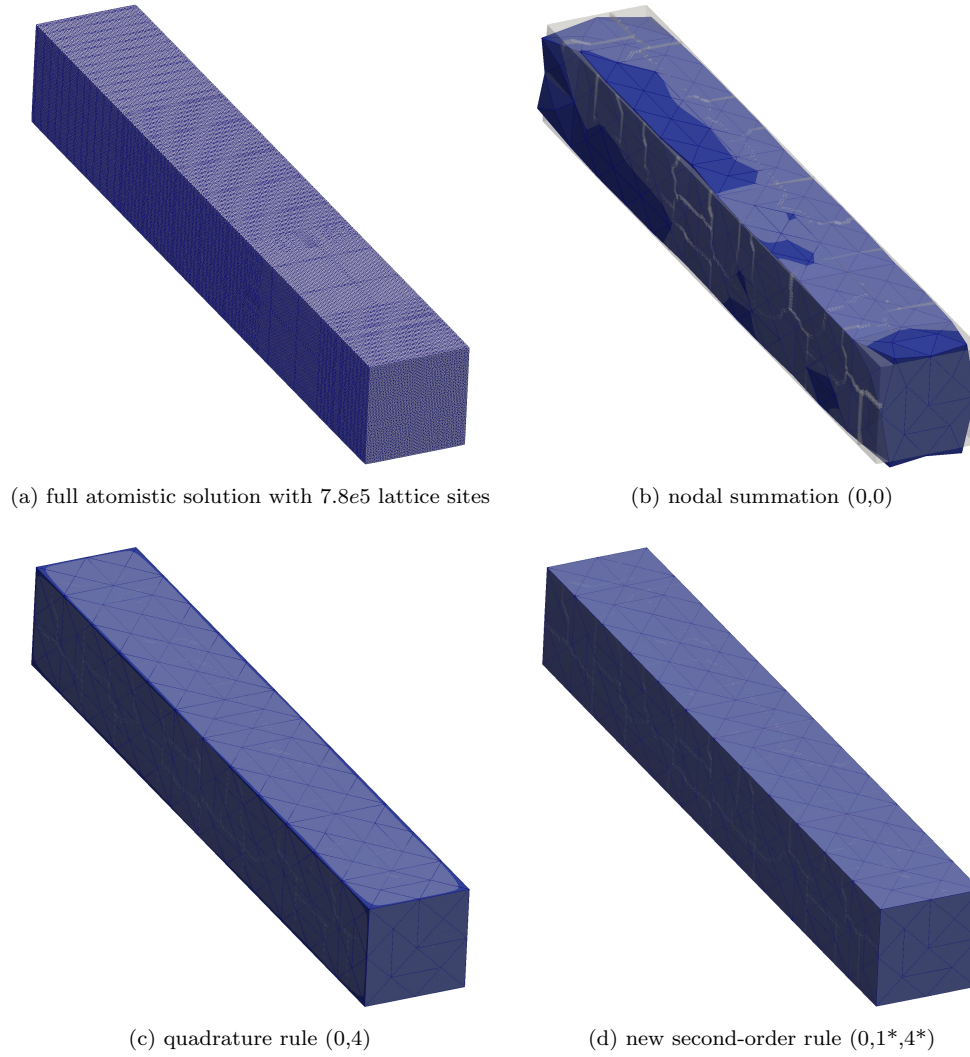


Figure 3.35: Deformed Cu nano-rods under uniaxial extension: (a) full atomistic solution, and (b)-(d) approximate QC solutions (the dark regions highlight deviations from the exact atomistic result). Particularly the nodal rule shows evidence of large errors due to force artifacts.

side lengths on the surface amount to four lattice spacings). For small element sizes (Figure 3.34a), the impact of the various summation rules is small and the stress-strain curves obtained from QC agree well with that obtained from lattice statics (which is also included in each of the three plots). As the element size increases, the summation rules gain importance. The new second-order summation rule is by far the most accurate, showing a response that is almost identical to the exact atomistic one. Most importantly, this is true for arbitrary coarsening ratios, whereas the errors produced by all other summation rules grow considerably with element size (the next best rule being four-point quadrature in this comparison).

Figure 3.35 shows the deformed nano-rods for different summation rules and for $L = 16a_0$ (the

rod has cross-sectional side lengths of 11.6 nm). The graphics show both the exact solution (light) and the QC solution (dark blue), thus indicating the errors in the deformed shape of the rod. Large errors arise from the nodal summation rule, where spurious force artifacts lead to a severely-deformed surface with artificial roughness. Again, the new second-order summation rule reproduces the deformed configuration in convincing agreement with the exact solution.

Evaluating the stress-strain response about the undeformed state yields the effective (structural) Young's modulus; i.e., we compute the slope of stress-strain curves such as those in Fig. 3.34 at zero strain. Results are plotted in Fig. 3.36a vs. the cross-section of the single-crystalline Cu nano-rod. In the limit of large diameters, the uniaxial modulus in the (100) direction agrees well with the expected value for a linear elastic material with cubic symmetry. The elastic constants can be determined from the interatomic potential via the Cauchy-Born rule; see for example [Tadmor et al., 1996], which yields $C_{11} = 168.4$ GPa, $C_{12} = 121.4$ GPa, and $C_{44} = 75.4$ GPa in agreement with experiments [Kittel, 1996]. Young's modulus for uniaxial tension-compression then follows as

$$E = \frac{(C_{11} - C_{12})(C_{11} + 2C_{12})}{C_{11} + C_{12}} = 66.69 \text{ GPa} = 0.417 \text{ eV/\AA}^3 \quad (3.1)$$

in agreement with [Simmons and Wang, 1971]. As becomes apparent from Figure 3.36a, this value is approached correctly in the limit of large rod cross-sections. Figure 3.36b illustrate the error introduced by the various summation rules, which confirms the general trends observed before with the new second-order summation rule showing errors of less than 2.5% throughout.

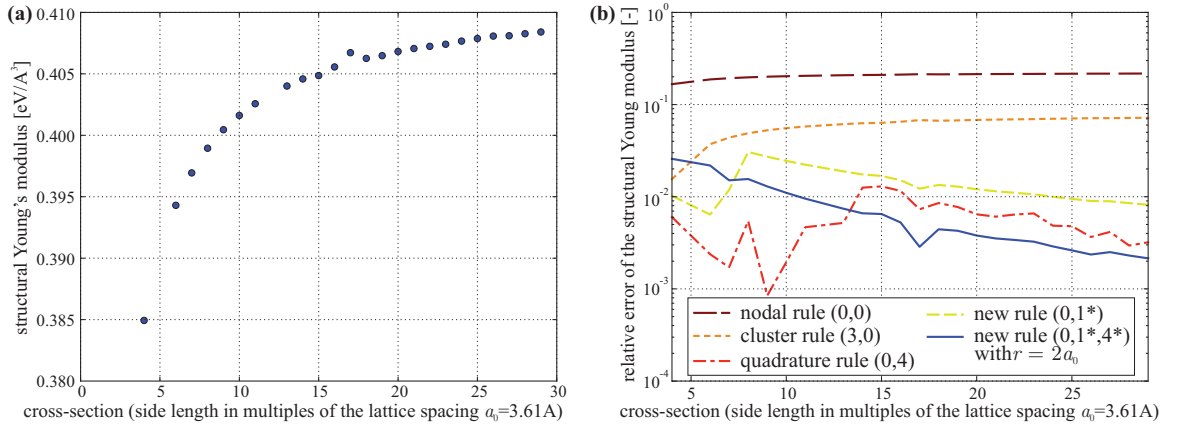


Figure 3.36: Size effects of the elastic modulus of a nano-rod: (a) elastic modulus vs. side length of the square cross-section of the nano-rod under uniaxial tension-compression; (b) error of the elastic modulus obtained from a QC representation of the nano-rod using various summation rules.

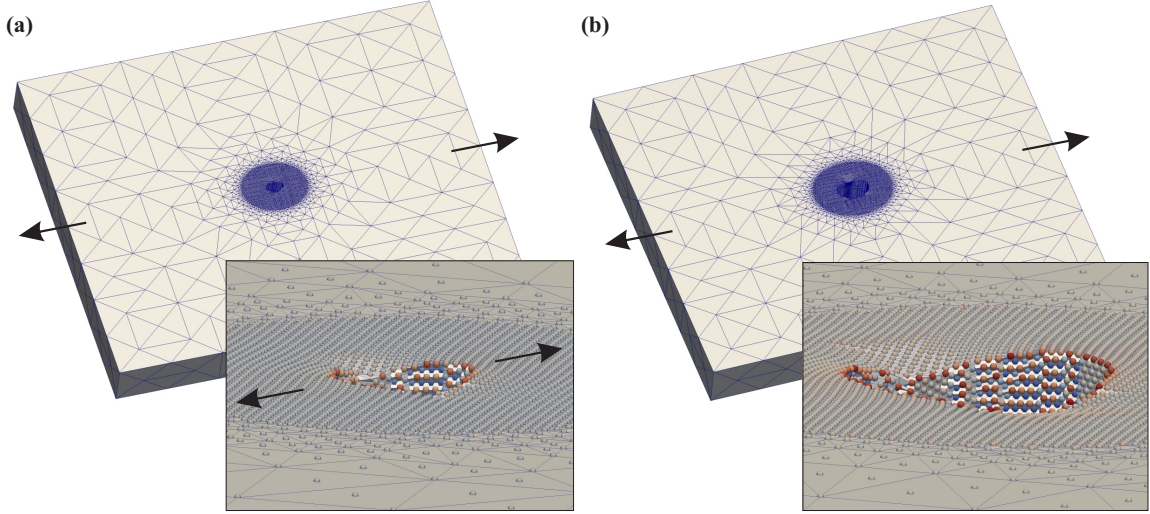


Figure 3.37: QC models of a thin plate with a cylindrical hole of (a) 3 nm and (b) 6 nm in diameter with zoomed graphics showing the deformation of the surface-nearest atomic sites in the vicinity of the hole at approximately (a) 2.8% and (b) 3.3% uniaxial strain (immediately before failure). Near the hole is full atomistic resolution with coarse-graining away from the center (arrows indicate the tensile loading direction).

3.5.3 Plate with a cylindrical hole

Finally, let us use the same QC methodology to model the failure of a thin Cu plate with an initially cylindrical hole, loaded in uniaxial extension in a (100) direction. The single-crystalline plate, shown in Figure 3.37, has extensions of approximately $92.4 \text{ nm} \times 92.4 \text{ nm} \times 11.6 \text{ nm}$ with a cylindrical hole at its center whose diameter we vary from about 3 nm to 6 nm. The region immediately surrounding the hole is modeled by full atomistics while the remainder of the plate is efficiently coarse-grained (of course, this is only a representative example – the general technique is applicable to, in principle, arbitrary system sizes).

We model the same scenario with a continuum description using the finite element method to obtain an approximate solution of the stress and strain distribution inside the plate with a cylindrical hole of the same dimensions. In order to mimic the atomistic scenario, we use a St. Venant-Kirchhoff hyperelastic energy density whose modulus tensor components \mathbb{C}_{ijkl} obey cubic symmetry with the stiffness constants obtained in Section 3.5.2 from the interatomic potential of [Dai et al., 2006] (which is used in the QC simulations). Of course, the continuum description is independent of the problem size and the ratio of the hole diameter to the plate thickness plays only a minor role for the chosen geometry. Figure 3.38 summarizes results obtained from the continuum study, including the out-of-plane deformation of the surface near the hole (matching quite well the atomistic/QC results of Figure 3.37) as well as the von Mises-stress distribution in the vicinity of the cylindrical hole with large stress concentrations at the hole on the axis perpendicular to the loading direction, as can be expected theoretically.

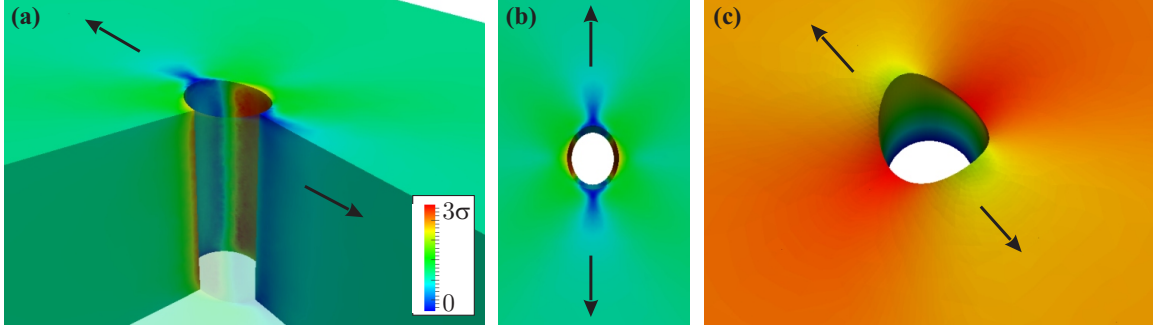


Figure 3.38: Finite element results for uniaxial loading at 2.4% strain, demonstrating the von Mises stress distribution (a) in the vicinity of the hole in three dimensions (including a cut through the sample) and (b) in a top view. The legend shows the stress normalized by the applied remote loading. Stress concentrations appear where microstructure is observed to form in Fig. 3.39; (c) illustrates the surface displacements near the hole (with out-of-plane displacements magnified by a factor of 40 for clarity and colors on the same color legend normalized by the maximum displacement).

When loading the coarse-grained atomistic simulations of Figure 3.37 beyond the elastic limit, microstructural defects form near those locations of highest von Mises stress. Figure 3.39 shows the formation of incipient dislocation loops spreading from the cylindrical hole and from the surface of the plate before the plate fails through cleavage (at zero temperature the dominant mode is brittle failure). Movies of microstructure formation for various radii may be found [here](#) for 1.5nm, [here](#) for 3.0nm, [here](#) for 4.5nm, and a synchronized movie containing all three is [here](#). Quite clearly, the surface plays an important role by affecting the stress state near the surface and by nucleating defects from and near the surface. The result is a size effect: while the plate with a 3 nm hole fails near 2.8% applied strain, the 6 nm hole fails not until 3.3% applied strain. The trend of failure strain as a function of hole radius can be seen in Figure 3.40.

Of course, the simulation of the hole and its immediate vicinity could easily have been carried out using MD due to the small hole diameter. However, the long-range elastic fields resulting from the hole and, in particular, due to microstructural defect formation require an appropriate representation far away from the cylindrical hole. Here, the QC methodology is a powerful tool that enables the modeling of, in principle, arbitrarily large systems containing atomistic features through coarse-graining; and unlike all previous schemes, free surfaces can be accounted for successfully by the presented new summation rules of second order.

3.6 Summary and conclusions

We have applied the fully-nonlocal energy-based quasicontinuum method to nanoscale structures including nano-particles, nano-rods, and plates with nano-sized holes, with and without free surfaces, to investigate the elastic and plastic material response. The intent of these investigations was to compare the various summation rules for the energy-based fully-nonlocal QC method and investigate

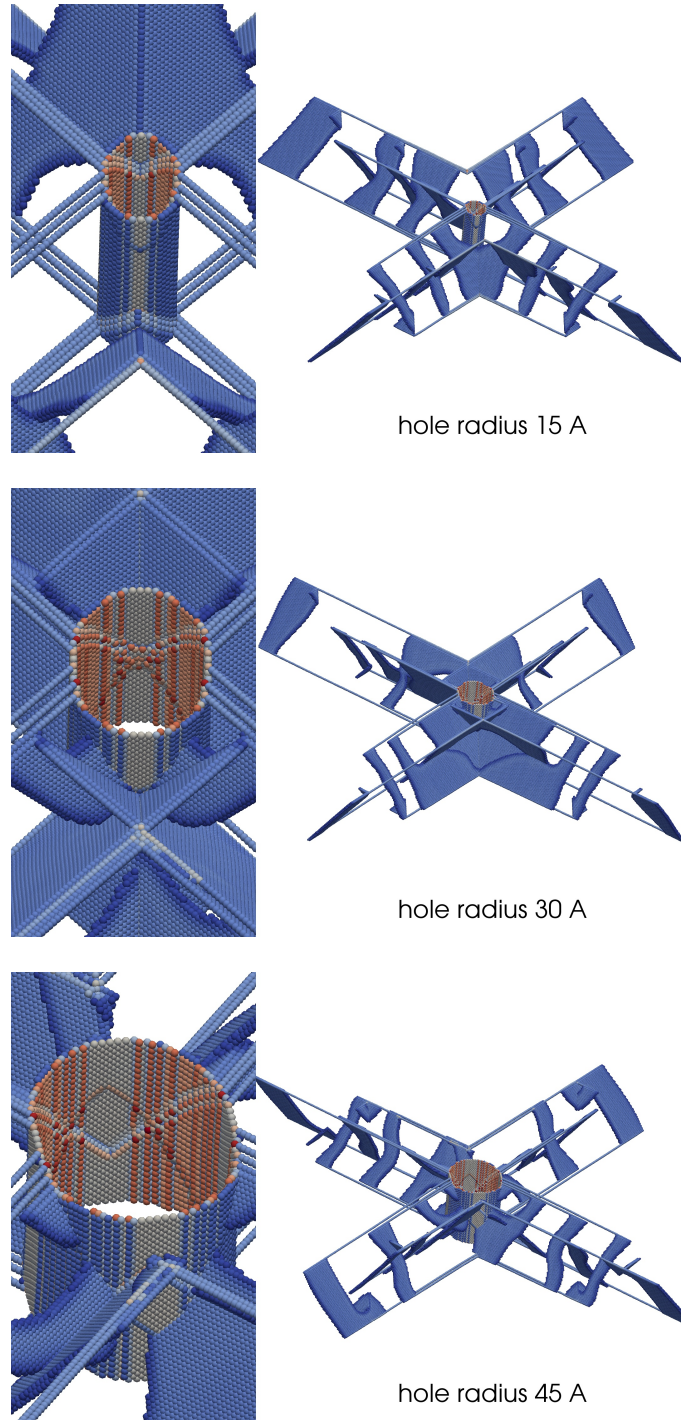


Figure 3.39: Formation of dislocation loops and surface defects around the cylindrical hole of (a) 1.5 nm, (b) 3 nm, and (c) 4.5 nm radii, loaded under uniaxial tension (defects have been identified and color-coded by the centrosymmetry parameter [Kelchner et al. \[1998\]](#); arrows indicate the loading direction). Movies of the individual simulations are available: [1.5nm](#), [3.0nm](#), [4.5nm](#), and a synchronized movie showing all three is [here](#).

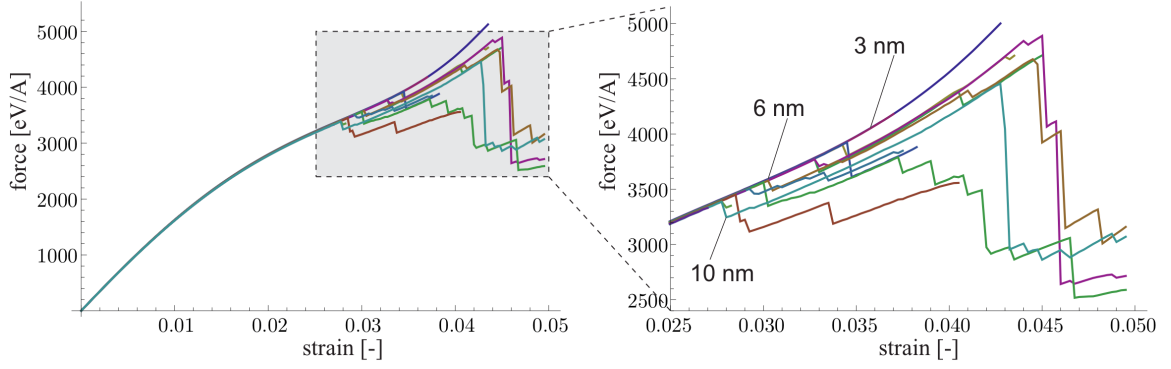


Figure 3.40: Reaction force for the scenario of a nanofilm with a cylindrical hole. The strain at initial dislocation emission is shown to be a function of the hole radius, a size effect.

their impact on the accuracy of results and the efficiency of simulations. Besides traditional nodal, cluster, and quadrature summation rules, we compared the first- and second-order realizations of a new summation rule based on sampling atoms with optimal weights.

- We first investigated examples of isolated, particular configurations (i.e., without equilibration). By looking at the energy approximation errors in a sample with multiple sharp and diffuse mesh interfaces, it was found that the new summation rules far outperformed the traditional summation rules, obtaining more than an order of magnitude reduction in energy error in many cases with no greater cost than quadrature rules.
- Continuing to investigate examples of a single configuration and because residual forces are commonly cited as a large problem in QC formulations, we then investigated the presence and magnitude of residual forces in two- and three-dimensional sharp and diffuse mesh interfaces. As was expected from the analysis in Chapter 2, traditional summation rules all exhibit ghost forces that scale with element size but the new summation rules show no spurious forces in the small- and large-element limits. The spurious forces produced by the summation rules are limited to a small range of element sizes where meshes transition out of full atomistic resolution.
- After examining isolated configurations, we turned our attention to integral simulations representing boundary value problems with external forces, boundary conditions, and equilibration. We first compared the QC solution to a geometrically-equivalent lattice statics solution for a nanovoid in a single crystal. Energy and displacement errors were quantified and reported, and the new summation rules produced astonishingly low error throughout almost the entire simulation domain in regular meshes with sharp mesh interfaces *as well as* randomized meshes with diffuse mesh interfaces. Not only did this confirm the performance of the new summation rules, but also confirmed the capability of the entire QC formulation to transmit remote loads

through coarse regions.

- To verify that the QC formulation could reproduce the macroscopic material constants to which the underlying potential was fitted, simulations were performed to calculate elastic constants from the curvature of the energy landscape. In all but a very pathological range of repeat atom densities, the new summation rule produced errors so small that they could not be visibly drawn on the same scale as the other summation rules.
- We then studied the classical example of nanoindentation in two and three dimensions. The mean force-displacement curve from many QC realizations and qualitative microstructure formation (quantitative comparison is not possible due to noise) was found to match well against the lattice statics solution. In three dimensions, a simulation of unprecedented size (16 million degrees of freedom) was performed to study microstructure formation.
- Lastly, several examples with surface effects were studied because of the new summation rule's unique ability to capture the energetic contributions of atoms near a surface. These examples included the calculation of surface energy, elastic moduli of nano-rods where the abundance of free surface affects the Young's modulus, and a plate with a cylindrical hole. In all examples, the new summation rule was able to capture surface effects better than any other rule.

The new scheme minimizes spurious and residual force artifacts as well as energy approximation errors compared to all other summation rules investigated here. This was demonstrated by instructive two- and three-dimensional examples using a variety of metallic interatomic potentials. In all examples, the comparison of QC results to those of full atomistics show convincing qualitative and quantitative agreement and, in particular, the superiority of the new second-order summation rule over the traditional QC summation rules (especially compared to the classical node-based cluster summation rules). Going beyond all previous QC techniques, the present multiscale modeling approach enables us to investigate the mechanics and physics of nanoscale systems efficiently through coarse-grained atomistics without the need to introduce ad-hoc corrections for free surfaces. Instead, the chosen approach naturally accounts for surface effects and enables the study of size effects in nanoscale metals.

With regards to simulations involving free surfaces, we would like to point out that most difficulties discussed here could alternatively be resolved by having full atomistic resolution across the entire surface, and this strategy has indeed been pursued in the scientific literature. However, full atomistic resolution on all surfaces implies high computational expenses, which severely limits the domain size to be modeled. If interesting phenomena can be expected to happen below the surface (e.g., defect interactions in nano-rods, whiskers, or nano-trusses) or if interesting surface effects appear localized on small fractions of the entire surface of a sample (such as growing voids or cracks

reaching the surface), then full atomistic resolution on all surfaces of the sample from the outset is a computational overkill (in particular in small-scale structures with an abundance of free surfaces). This is exactly where coarse-graining techniques such as the ones presented here gain importance, since they allow us to restrict full atomistic resolution to where it is required.

Chapter 4

Implementation

4.1 General overview

4.1.1 Scope

In order to use the formulation of Chapter 2 to perform calculations on systems of engineering and scientific interest, the following features of an implementation are necessary:

- 2D and 3D geometry
- Support for the framework of sampling atoms through which existing and novel summation rules are expressible
- Strong-scaling single node performance, allowing for simulations with tens or hundreds of thousands of degrees of freedom on a single computer
- Weak-scaling parallel performance, allowing for simulations with tens of millions of degrees of freedom across many computers

Existing implementations Some prominent implementations of the QC method already exist, but do not meet our requirements:

- There exists a freely-available 2D QC code produced and maintained by Miller and Tadmor [[Miller and Tadmor, 2011](#)] which is 2.5D, zero temperature, and implements the local/non-local QC formulation. Various extensions of this code exist (e.g., finite temperature [[Dupuy et al., 2005](#)]), and a 3D implementation is in progress.
- Knap and Marian et al. developed a 3D implementation which is owned by Lawrence Livermore National Lab, supports zero-temperature with a finite-temperature extension, and is based on the nonlocal force-based formulation. It is not freely-available.

- Ariza, Romero, and Ortiz et al. use a 3D implementation of “hot-QC” [Venturini et al., 2014], an energy-based formulation with cluster summation rules.

Though several two-dimensional QC implementations exist, none provided the extensibility required for sampling atoms nor the performance required for the large, efficient simulations targeted by this work. Because of this lack, a new QC realization was designed and implemented specifically for the purpose of supporting the above features. The initial intended use of the new implementation was quasistatic zero-temperature simulations, while allowing for future extensions in both of the areas of dynamics and temperature.

4.1.2 Outline and data structures

A simulation is a series of configurations of boundary conditions and external forces, called “load-steps” in this section. At each loadstep the repatom positions are equilibrated, after which boundary condition and external force reactions are measured. The general outline of the program is shown in Listing 4.1:

Listing 4.1: General outline

```

initialize repatom (geometric) data structures
perform parallel decomposition of the domain and geometric data structures
initialize sampling atom data structures
for every loadstep
  update loads (boundary conditions and external forces)
  equilibrate repatom positions

```

Though in principle any solver may be used to equilibrate repatom positions, solvers which require stiffness calculations are prohibitively expensive for the large-scale calculations in this thesis, and we restrict our attention to iterative solvers. In particular, the inertial relaxation algorithm FIRE of Bitzek et al. [2006] was used for all examples shown in Section 3.

To understand the core data structures and program flow, we take the example of calculating repatom forces for the embedded atom method (see Section 2.1.4) where the force on a given repatom k is as shown in Equation 4.1:

$$\tilde{\mathbf{F}}_k(\mathbf{x}) = - \sum_{\alpha=1}^{N_s} w_{\alpha} \sum_{j \in n_I(\alpha)} \left[\frac{1}{2} \Phi'(r_{\alpha j}^h) + \mathcal{F}'(\rho_{\alpha}^h) f'(r_{\alpha j}^h) \right] \frac{\mathbf{r}_{\alpha j}^h}{r_{\alpha j}^h} [N_k(\mathbf{X}_{\alpha}) - N_k(\mathbf{X}_j)], \quad (4.1)$$

In order to compute these forces the following key data structures are necessary:

- A collection of sampling atoms, each of which has a weight (to provide w_α), a deformed position (to provide $r_{\alpha j}^h$), and a set of shape function values (to provide $N_k(\mathbf{X}_\alpha)$).
- Undeformed repatom positions and a mesh connecting them for determining shape function values, as well as deformed repatom positions for interpolating deformed positions. The role of and difficulties associated with the mesh are discussed in Section 4.3.1.
- A collection which stores the lattice sites which are the “neighbors” of each sampling atom. These are lattice sites which are within the potential’s cutoff radius to each sampling atom *in the deformed configuration*, such that the distance ($r_{\alpha j}^h$) produces nonzero values of $\Phi'(r_{\alpha j}^h)$ or $f'(r_{\alpha j}^h)$. Each lattice site is represented as a tuple of shape function values (which provide $N_k(\mathbf{X}_j)$), from which deformed positions and distances can be computed (which provide $r_{\alpha j}^h$). The lattice sites are stored in a `NeighborhoodContainer` which is described in Section A.

Listing 4.2: Force calculation pseudocode

```

for every sampling atom
  calculate this sampling atom’s  $\rho_\alpha^h$ 
  set sampleAtomsForce = zeroVector
  for every neighbor lattice site in the deformed configuration
    calculate  $r_{\alpha j}^h$ 
    calculate forceScalar =  $0.5 * \Phi'(r_{\alpha j}^h) + \mathcal{F}'(\rho_\alpha^h) f'(r_{\alpha j}^h)$ 
    for every repatom on which this lattice site depends
      set force = forceScalar * unitVector
      add force to sampleAtomsForce
      repatomForces(repatomIndex) +=
        force * sampleAtomWeight * shapeFunctionValue
    for every repatom on which this sampling atom depends
      repatomForces(repatomIndex) +=
        sampleAtomsForce * sampleAtomWeight * shapeFunctionValue
  add external forces to all repatoms

```

4.2 Computing forces and equilibrating

Though equation 4.1 describes the force on a single repatom k as the summation over all sampling atoms, because of the multiplicative shape function factors, only the sampling atoms which have nonzero shape function values for repatom k have any effect and the force calculation is better expressed as an algorithm performed over the sampling atoms. With this transformation, the force

calculation becomes a loop over sampling atoms and an inner loop over sampling atom neighbors (see Listing 4.2). The force on each neighbor is calculated and added to the repatoms whose shape function values are nonzero at the *neighbor's* (undeformed) position, which handles the summation with the $N_k(\mathbf{X}_j)$ term. Each neighbor's force is summed and then also added to the repatoms whose shape function values are nonzero at the *sampling atom's* (undeformed) position, which handles the summation with the $N_k(\mathbf{X}_\alpha)$ term.

Note that calculating forces on a per-sampling atom operation instead of a per-repatom operation brings as a consequence coordination among the parallel threads and processes. If the parallelization is performed on the level of the sampling atoms, multiple threads can be writing to a given repatom's force at a time and race conditions will corrupt results.

Listing 4.3: Time integration scheme

```
Derive  $\mathbf{f}^i$  from  $\mathbf{x}^i$ 
Set  $\mathbf{a}^i = \mathbf{f}^i / \mathbf{m}$ 
Calculate  $\mathbf{v}^i = \mathbf{v}^{i-1/2} + \frac{1}{2}\mathbf{a}^i \Delta t$ 
Adjust  $\mathbf{v}^i$  and  $\Delta t$  with FIRE algorithm
Calculate  $\mathbf{x}^{i+1} = \mathbf{x}^i + \mathbf{v}^i \Delta t + \frac{1}{2}\mathbf{a}^i \Delta t^2$ 
Calculate  $\mathbf{v}^{i+1/2} = \mathbf{v}^i + \frac{1}{2}\mathbf{a}^i \Delta t$ 
```

4.2.1 Solver

For molecular dynamics, the use of the Fast Inertial Relaxation Engine (FIRE) technique of [Bitzek et al. \[2006\]](#) shows significant improvement over the more commonly-used Polak Ribière or BFGS methods, as measured by number of necessary energy evaluations. FIRE is a modification of traditional MD trajectories with two core properties: velocities are reset when the system sees “uphill motion” (negative power) and velocities are calculated from a mixture of velocity and force vectors. Both the timestep and the mixture of velocity and force vectors (controlled by the FIRE parameter α) are dynamically-adaptive to accelerate progress towards energy wells.

FIRE lends itself well to our QC equilibrations because of the their computational resemblance to MD. On each loadstep, we employ the modified velocity Verlet time integration technique (shown in Listing 4.3) with FIRE modifications to velocity and timestep. $\mathbf{x}, \mathbf{v}, \mathbf{a}, \mathbf{f}, \mathbf{m}$ represent repatom positions, velocities, accelerations, forces, and masses (respectively) and superscript represents the timestep index. As is common practice with velocity Verlet, the velocity “half-update” is denoted by a half timestep, though the designation is only conceptual and only one copy of the velocities exists per timestep.

To model dynamics, the above has been extended to include inertia defined through the repatom

weights and a central difference scheme of second order on the acceleration of repatoms. That is, instead of solving for $\mathbf{f} = 0$, we solve

$$\mathbf{f}_\alpha^{i+1} = M_\alpha \frac{\mathbf{x}_\alpha^{i+1} - 2\mathbf{x}_\alpha^i + \mathbf{x}_\alpha^{i-1}}{\Delta t^2}, \quad (4.2)$$

where α refers to a particular repatom and M_α is the repatom’s weight. This inertial term is converted to an external force via the Lagrange-d’Alembert principle.

A movie illustrating the repatom velocities for a quasistatic simulation of uniaxial extension can be found [here](#). A movie of the repatom velocities produced by implicit dynamics on the same scenario is available [here](#). In the quasistatic simulation, much larger changes are made to the boundary conditions and external forces between loadsteps, as visible in the left panel of the movie. In simulations that use implicit dynamics with high strain rates, many more loadsteps are performed and the effects of applied loads do not propagate through the entire domain at once. In particular, the time scale at which dislocations and other local phenomena move can be restricted, as is seen in the slow movement of the dislocations formed at the end of the movie. The behavior of that same domain under shear deformation can be seen [here](#).

Note that computing the forces is done (non-traditionally) as the first step on each timestep so that forces do not have to be valid between FIRE iterations, to allow for easier program structure with adaptive model refinement. In total, the expanded algorithm in QC vocabulary and including communication is shown in Listing 4.5.

4.2.2 Repatom weights

A difference between the traditional MD velocity Verlet time integration scheme and the one presented in Listing 4.3 is the “masses” of each degree of freedom, called repatom *weights* in this work, which are not the same for all degrees of freedom as they are for MD. In the current realization of the algorithm, the computation of repatom weights is dependent upon the choice of summation rule. Note that these repatom weights are unrelated to the sampling atom weights used in summation rules (see Section 2.1.2).

Listing 4.4: Repatom weight calculation for optimal summation rules

```

for each element  $e$ 
   $volume = \text{calculate volume of } e$ 
  for each vertex  $v$  of element  $e$ 
     $\text{repatomWeights}(v) += volume / (\text{SpatialDimension} + 1)$ 

```

For quadrature and cluster summation rules of the form (n_{Cl}, n_Q) , repatom weights are calculated by tessellation just as sampling atom weights are. However, 3D tessellation is difficult to use for domains of any geometry other than axis-aligned boxes or cylinders (such as the pillar in Figure 4.6). One of the advantages of the new, optimal summation rules $(0, 1^*)$ and $(0, 1^*, 4^*)$ is that they do not need global tessellations and work well on meshes of arbitrary shape. Therefore, for optimal summation rules repatom weights are not calculated via tessellation but by equally distributing the volume of each element to the repatoms at its vertices, the barycentric dual volume. That is, the repatom weight is calculated as shown in Listing 4.4:

4.3 Key data structures

4.3.1 Mesh

Because the presented QC formulation uses linear interpolation to calculate sampling atom and neighbor positions, each simulation requires a simplicial mesh that covers the convex hull of and connects the undeformed repatom positions. The mesh is an essential component of the simulation and many operations use it, including determining sampling atom positions, finding new repatom positions for adaptive model refinement, defining neighbors of sampling atoms, and parallel domain decomposition. In the current realization of the QC formulation, each different scenario populates undeformed repatom positions, which are then given to an external mesher to form a mesh that covers the convex hull of the input points.

The quality of the mesh can affect the QC solution in two ways: local orientations of elements can change force calculations and degenerate elements cause many problems, including force artifacts. As shown in Figure 4.1, for surface calculations the local orientations of mesh elements can lead to non-uniform forces even on meshes with uniform repatom spacing and distribution. The initial undeformed repatom positions as well as the externally-created mesh are shown in panel (a). As shown in panels (e) and (f), because the mesh is not uniform repatoms have different numbers of incident elements and different numbers and configurations of sampling atom neighbors that contribute forces to the repatom. This non-uniformity in the mesh leads to the non-uniform force distribution of panel (d), which leads to the wrinkling of the equilibrated surface shown in panel (b). If the meshing is done by hand such that all repatoms “see” the same arrangement of incident elements, the same simulation generates the uniform residual forces of panel (c) and does not lead to wrinkling in the equilibrated surface. This type of incorrect force only occurs at free surfaces and is not created in the bulk of simulation domains, even when the local orientations of mesh elements connected to a given repatom are not the same.

In two dimensions, meshing is straight-forward and many external meshing software packages function adequately. Meshing in three dimensions is more difficult, primarily because the unde-

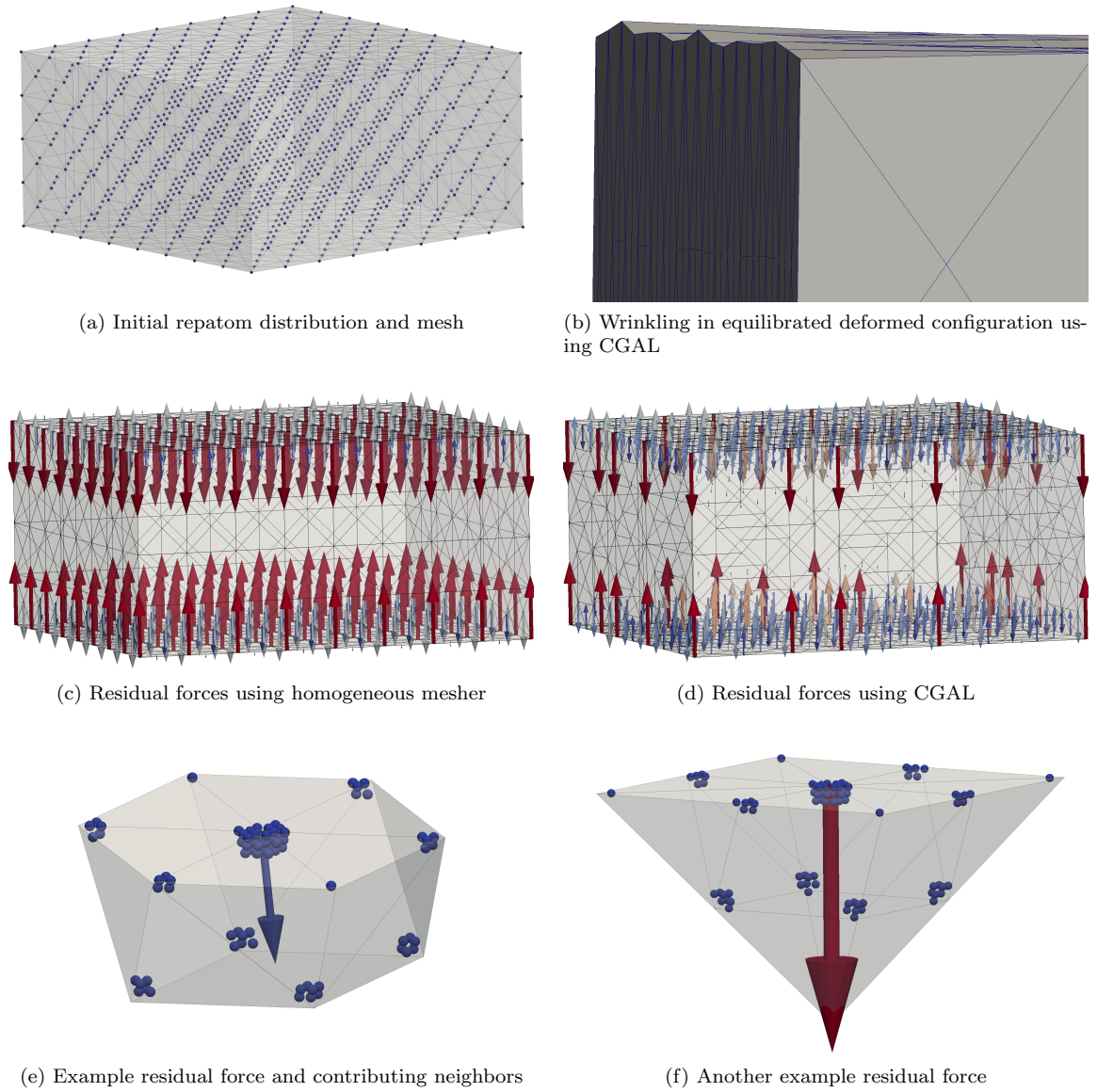


Figure 4.1: Mesh dependence in surface calculations.

formed repatom positions given to the meshers contain many instances where subsets of points are co-spherical, an input configuration that is known to be difficult for triangulation. Unlike inhomogeneous local element orientations which only create nonuniform forces in the presence of free surfaces, degenerate elements produced by external 3D meshers pose a serious challenge, causing difficulties in sampling atom population, adaptive model refinement, and residual forces *even in the bulk* of simulation domains. All 3D meshers we tested (Vtk [Shroeder et al., 2006], hull [Clarkson et al., 1993], Qhull [Barber et al., 1996], Tetgen [Si., 2015], and CGAL [Hert and Seel, 2015]) create degenerate elements to varying degrees.

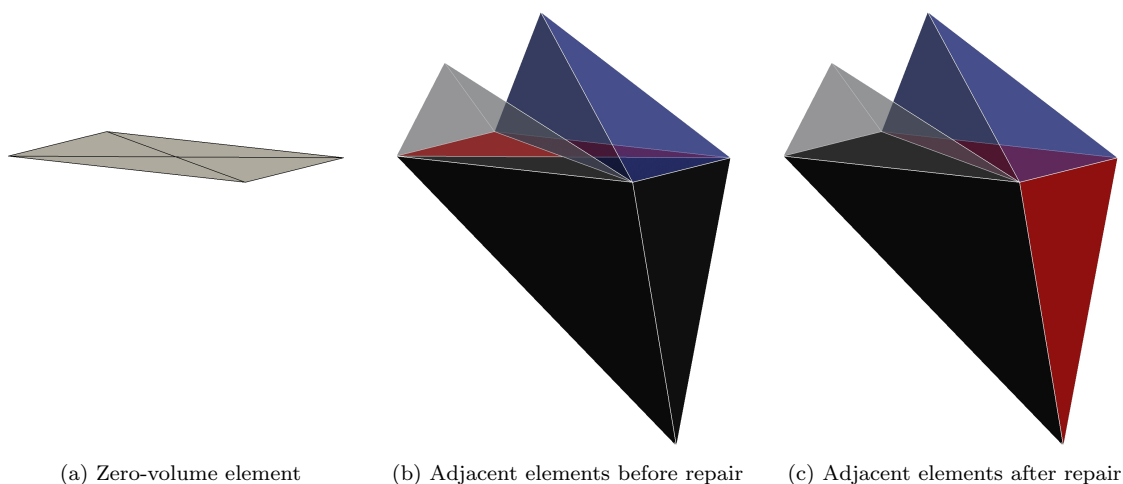


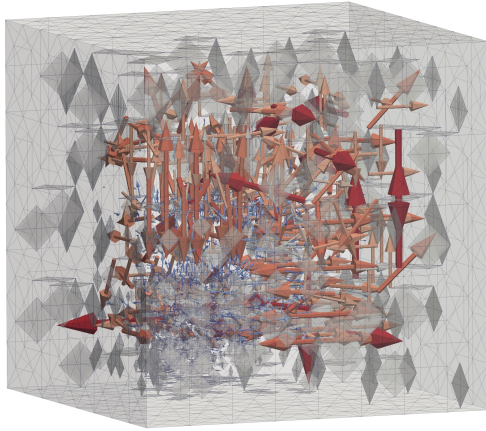
Figure 4.2: Zero-volume element repair.

CGAL is used in the current implementation, after which we use a custom repairing process to remove as many degenerate elements as possible. Figure 4.2 illustrates the repairing process (sliver removal) on a single zero-volume degenerate tetrahedron, where the four elements adjacent to the zero-volume tetrahedron are reconnected in a way that eliminates the zero-volume element with a 3-2 edge flip. Isolated zero-volume elements (which form planar quadrilaterals) account for the vast majority of the degenerate elements in the meshes, though pairs of zero-volume elements (forming planar pentagons) are also found and removed (see Figure B.1 in Appendix B).

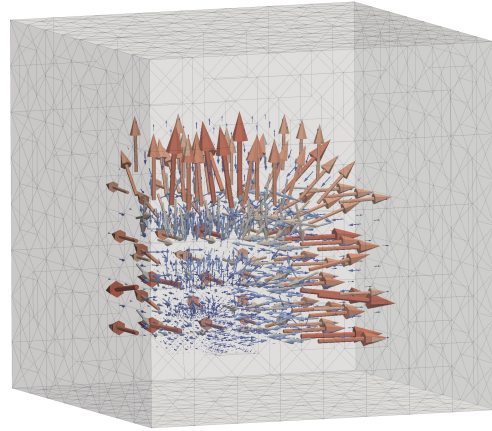
Though the repairing process does remove a significant fraction of the degenerate elements, it does not completely repair the mesh and the remaining degenerate elements complicate, reduce the efficacy of, or even substantially hinder many aspects of the simulation, including sampling atom generation for the new second-order rule and adaptive model refinement. Figures 4.3 (regular mesh) and 4.4 (irregular mesh) show the degenerate elements and resultant residual forces before and after repairing, for nodal summation, the new first-order rule, and the new second-order quadrature rule. As is shown in these two figures, the meshers create more zero-volume elements when the undeformed repatom positions are regularly arranged, as is the case for many structured QC meshes with sharp interfaces. A full comparison of the quality and performance of various 3D meshers as well as a discussion about the custom repairing process is presented in Section B.

4.3.1.1 Vacancies and nonconvexity

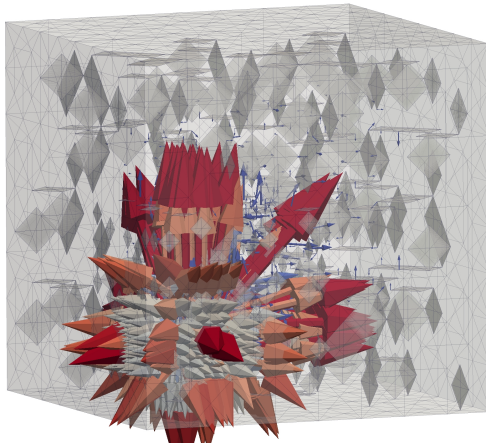
In order to support simulations having vacancies within otherwise-convex geometry, each scenario not only provides a collection of undeformed repatom positions but may optionally provide a collection of “vacant locations”, or lattice sites that will be excluded from the simulation. After meshing, any



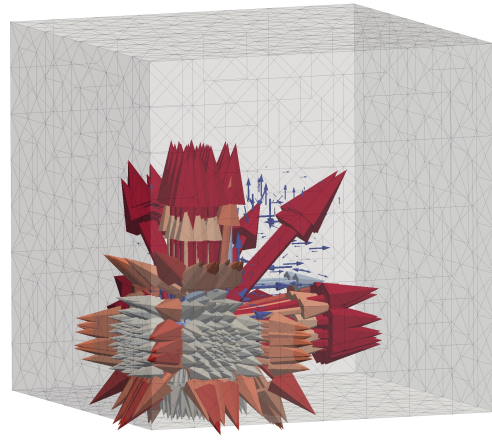
(a) Nodal summation, unrepaired mesh



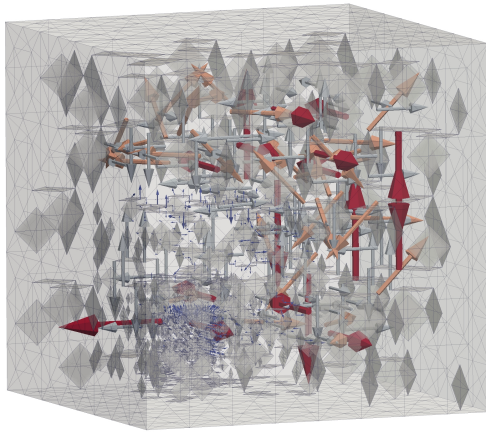
(b) Nodal summation, repaired mesh



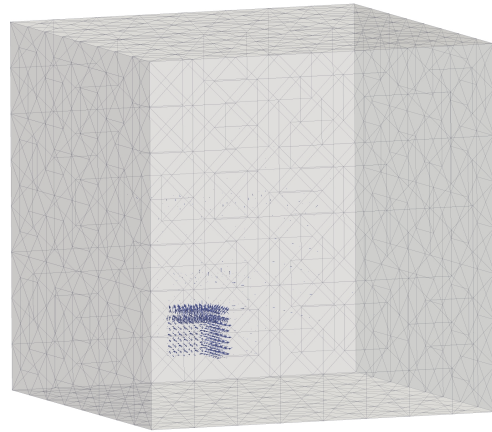
(c) New first-order summation, unrepaired mesh



(d) New first-order summation, repaired mesh

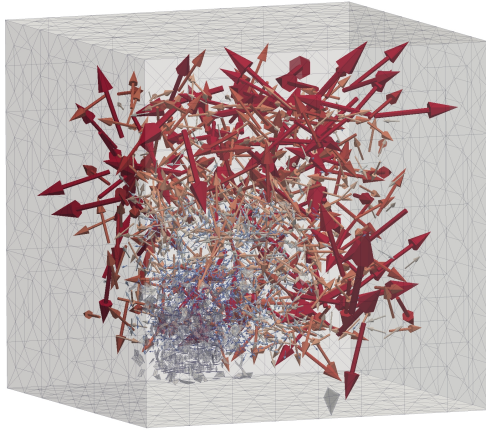


(e) New second-order summation, unrepaired mesh

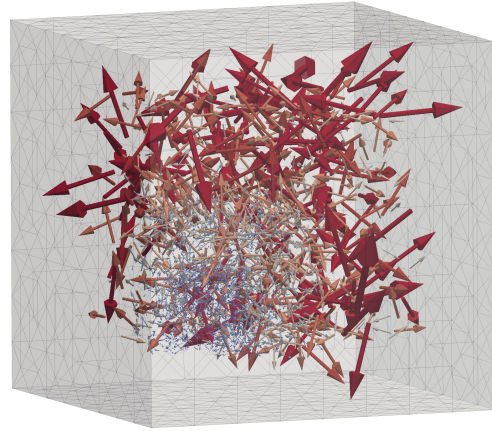


(f) New second-order summation, repaired mesh

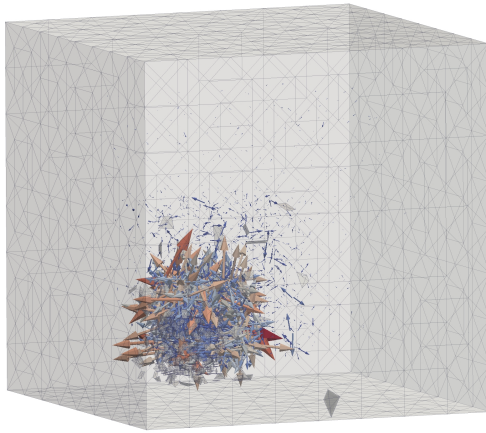
Figure 4.3: Degenerate elements and residual forces on a regular mesh. The scale of forces is the same for each summation rule, but not the same across summation rules.



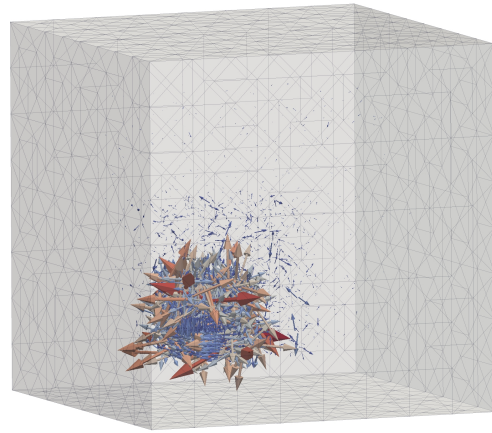
(a) Nodal summation, unrepaired mesh



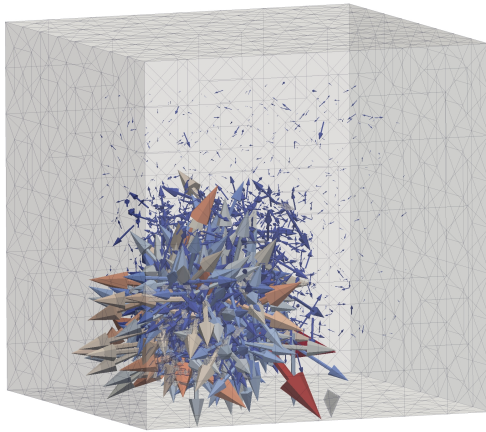
(b) Nodal summation, repaired mesh



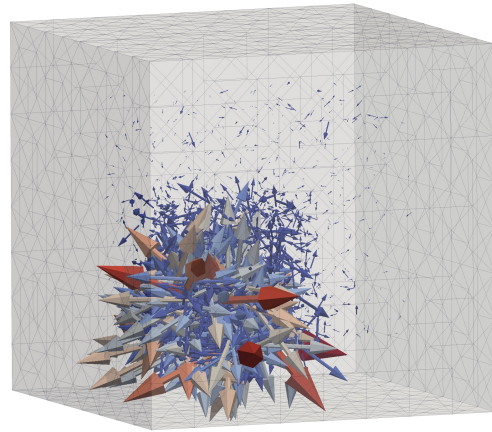
(c) New first-order summation, unrepaired mesh



(d) New first-order summation, repaired mesh



(e) New second-order summation, unrepaired mesh



(f) New second-order summation, repaired mesh

Figure 4.4: Degenerate elements and residual forces on an irregular mesh. The scale of forces is the same for each summation rule, but not the same across summation rules.

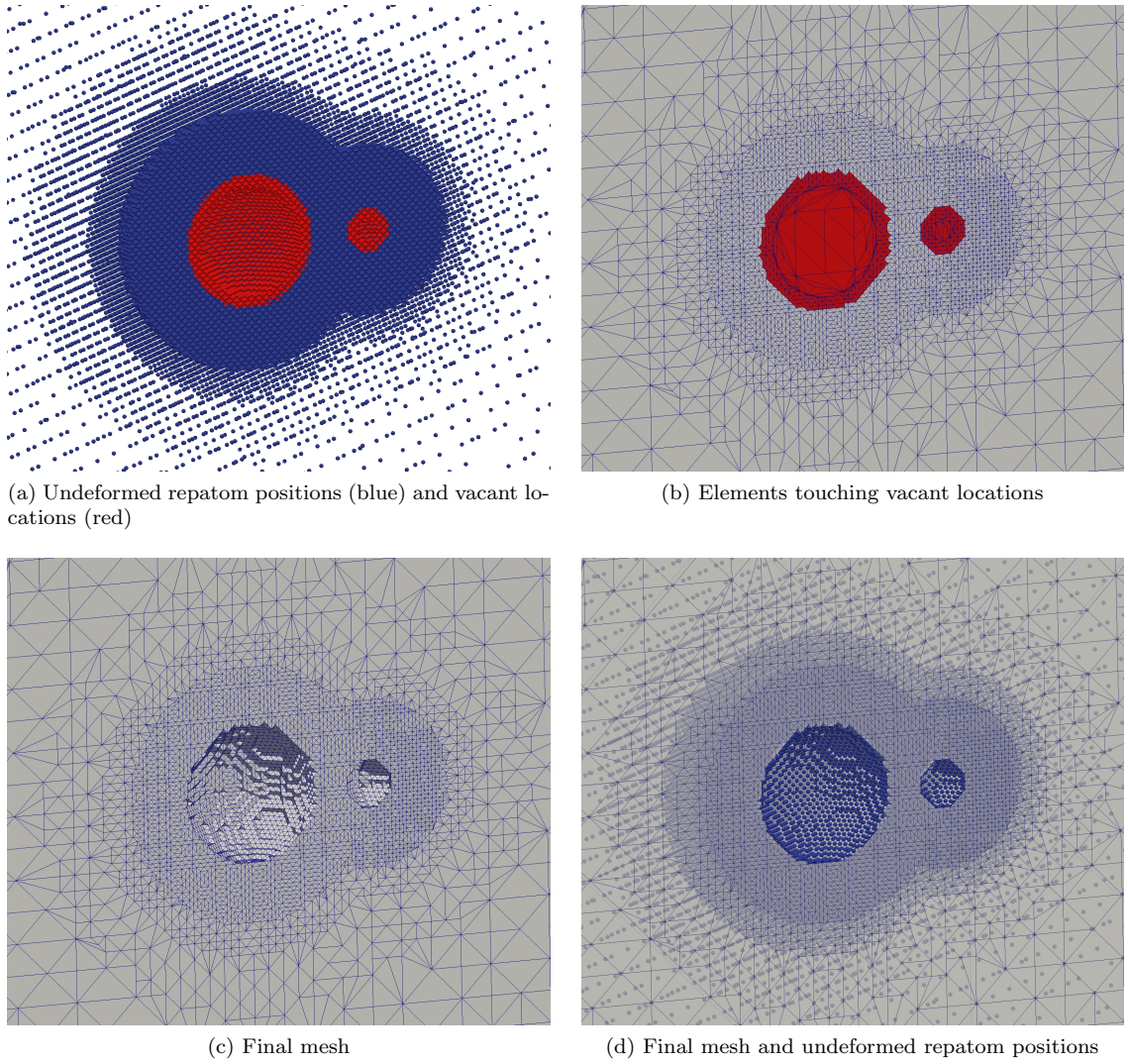


Figure 4.5: Producing voids and vacancies.

element touching one of the vacant locations is discarded, a process illustrated in Figure 4.5. Panel (a) shows the undeformed repatom positions in blue and the vacant locations in red, and the union of the two sets of points are meshed. In this example, only a few layers of vacant locations are kept on the inside of the voids, for efficiency. Panel (b) highlights in red the elements that touch vacant locations, which are removed to produce the final mesh shown in panels (c) and (d) (also showing original undeformed repatom positions through the semi-transparent mesh).

Vacant locations allow for holes within geometry and could be used to produce nonconvex bodies by surrounding the body with vacant locations, but this is cumbersome and expensive. In order to support simulations with nonconvex outer geometry, each scenario can also provide a utility object which is used to select elements to remove from the mesh. For example, to generate nanopillars as

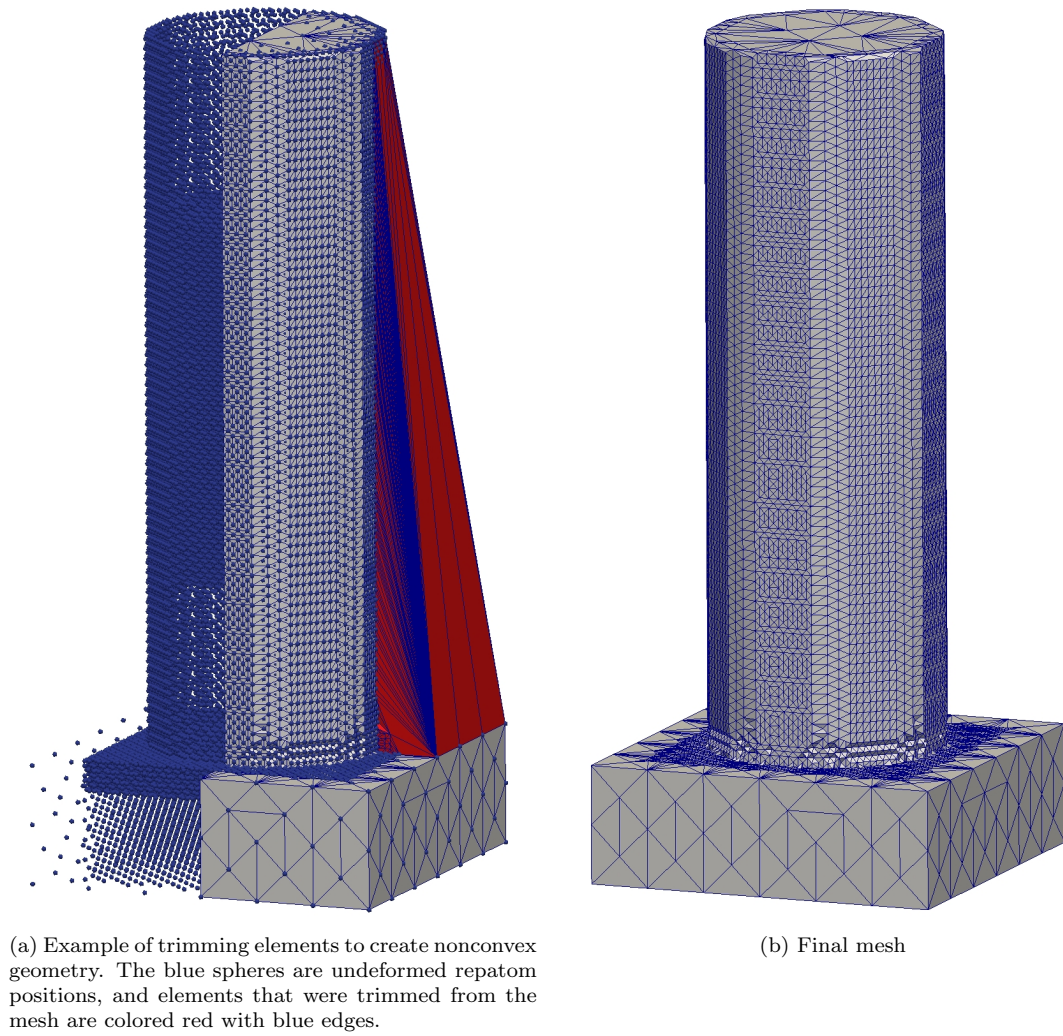


Figure 4.6: Producing nonconvex geometry.

shown in Figure 4.6, any elements with barycenters outside of the cylindrical pillar or the rectangular base are removed. A ring of vacant locations around the base of the pillar is used to prevent the formation of elements which extend from the base up to the pillar, outside of the intended geometry.

4.4 Distributed-memory parallelization

Due to heat and power limits, the execution speed of a single-threaded program on modern computers has essentially plateaued over the last decade. Unlike during the boom of the 1990s where increasing clock speeds meant increased execution speed simply by buying a new processor, software is no longer expected to run faster on newer processors without modification. Nevertheless, for at least the immediate future Moore's law is still marching forward (albeit more slowly) and computer

chips continue to have more transistors available. Though the execution speed of single-threaded programs is not increasing, the increased number of transistors from Moore’s law allows for multiple parallel threads of execution. In short, advances in computation allow for greater throughput via parallel execution but not faster serial execution, and software designed to be highly-performing must leverage modern parallel architectures.

In order to leverage the available throughput of modern processors, programs must first ensure that they are spending as little time waiting for memory access as possible. This is achieved via cache-aware data structures, algorithms, and memory access patterns, with judicious use of software prefetching. Next, software must utilize the vector registers available within a single computational core that allow for operations to work on more than one number at a time. Then, the program must be split into several independent execution threads so that all available cores on the processor can be used concurrently. Once these three steps are performed, the program can leverage the full capacity of a single processor.

Simulations that only use a single processor are ultimately limited in size and speed. Modern supercomputers are collections of compute nodes which are often very similar to consumer desktop computers. These compute nodes are connected via an interconnect so that they can communicate and work together. This type of computer is called a distributed-memory computer because the memory is distributed among many compute nodes. In order to perform larger simulations than what is possible on a single computer, programs can be written to pass messages between computers and can be run such that they concurrently use the many individual computers of a modern supercomputer. Depending on the strategy chosen to realize this type of distributed-memory parallelization, the modifications required to the program can be either minimal or extensive.

When running on a distributed-memory computer, many instances (copies) of the program are started, each on (possibly) different computers, and the instances coordinate and work together. The easier way to introduce this style of distributed-memory parallelization is to have every instance of the program contain the entire state of the simulation. In this model, the most computationally-intensive sections of the program are divided up among the different instances, then all instances share their updated information with all other instances. This model is easy to implement, but the realizable simulation size is limited because every instance of the program must contain the state of the entire simulation. For most modern supercomputers, this limits the number of repeatoms to the low tens of millions.

The formulation and implementation of this thesis targets simulations of more than tens of millions of repeatoms, and as such a different strategy must be used. In the current implementation, each instance of the program only has a subset of the domain, and no single instance contains the state of the entire simulation. This strategy significantly complicates the formulation but allows for simulations of arbitrary size. Figure 4.7 shows the decomposition of a nanoindentation simulation

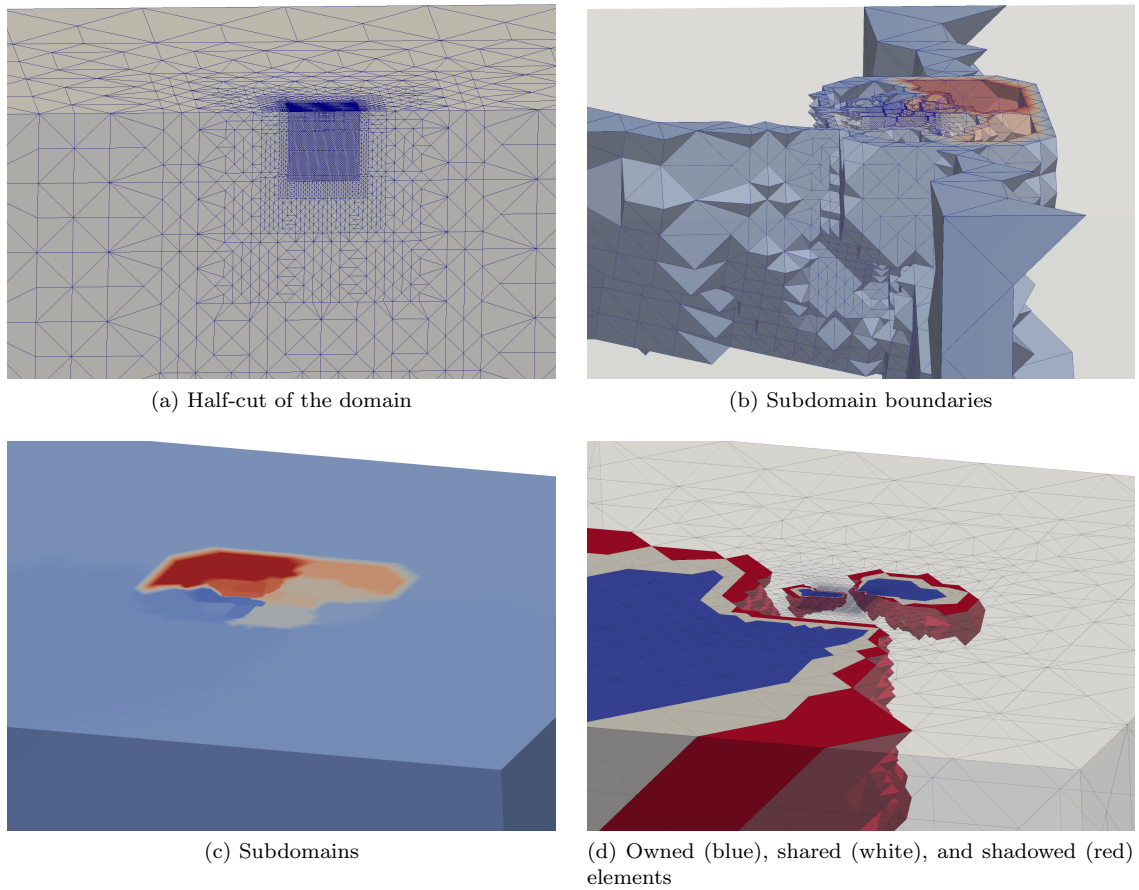


Figure 4.7: Nanoindentation domain decomposition summary.

for 60 instances. Panel (a) illustrates a half-cut of the simulation domain, with an atomistic core and mesh coarsening away from the core. Panel (b) highlights the boundaries of the subdomains of the instances, and panel (c) visualizes the subdomains themselves. Panel (d) demonstrates what each instance of the program actually sees: they own a portion of the domain (in blue), share some elements with other instances (in white), and have copies of some elements that belong to other instances (in red).

Because each instance of the program only has access to a portion of the domain, though any given repatom is uniquely owned by a single instance of the program the total force on that repatom is divided into contributions from many different instances and must be collected and combined. In order for the owner of the repatom to correctly update the repatom's position, the force contributions must all be sent to the owners of the repatoms. After contributions are collected, each instance updates the position of all owned repatoms. Finally, the owners of the repatoms must send their updated positions to all other instances that know of those repatoms. The interleaving of this communication into the solver's logic is shown in Listing 4.5.

Many more details about the distributed-memory formulation of this method are discussed in Appendix C, including how a simulation is decomposed and initialized, as well as how the adaptive neighborhoods of Appendix A function in the context of the parallel implementation.

Listing 4.5: Solver pseudocode with communication

```

while forces are not equilibrated
# calculate forces:
    update sampling atom and neighbors positions for current configuration
    compute repatom force contributions from all of this rank's sampling atoms
    send force contributions on un-owned repatoms to their owners
    receive force contributions on owned repatoms from other ranks
    apply boundary conditions to repatom forces
# do velocity Verlet update on all velocities:
    for each repatom
        repatomAcceleration = repatomForce / (mass * repatomWeight)
        repatomVelocity += 0.5 * repatomAcceleration *  $\Delta t$ 
    apply boundary conditions to repatom velocities
# perform FIRE algorithm:
    compute totalPower as the sum of all individual (owned) repatom forces
        dotted with their corresponding velocities
    adjust repatom velocities with  $\alpha$ 
    reduce totalPower sum across all ranks
    if (totalPower > 0)
        increase timestep, decrease  $\alpha$ 
    else
        decrease timestep, reset  $\alpha$ 
        set all repatom velocities to zero
# do velocity Verlet update on positions and velocities:
    for each repatom
        repatomAcceleration = repatomForce / (mass * repatomWeight)
        repatomPosition += repatomVelocity *  $\Delta t$  + 0.5 * repatomAcceleration *  $\Delta t^2$ 
        repatomVelocity += 0.5 * repatomAcceleration *  $\Delta t$ 
    apply boundary conditions to repatom positions and velocities
    send updated positions of owned repatoms of which other ranks have shadow copies
    receive updated positions of un-owned repatoms

```

4.5 Weights for optimal summation rules

One of the key components of the formulation described in Chapter 2 is the introduction of sampling atoms, or the separation of sampling energetic values from the repatoms used for kinematics by the use of summation rules. While previously-used summation rules of the nodal, cluster, and quadrature varieties all exhibit inaccuracies that lead to spurious forces and incorrect integration of energy, the new optimal summation rules produce errors that are orders of magnitude below those of previously-used summation rules with the same computational cost.

A summation rule is a prescription of where sampling atoms are placed (relative to either repatoms, elements, or both) as well as a method of calculating the weights of those sampling atoms. The optimal summation rule described in Sections 2.3.1.3 (1D), 2.3.2.2 (2D), and 2.4 (3D) places sampling atoms in specific locations where they will represent many atomic sites with similar neighborhoods. This new summation rule comes in zeroth-, first-, and second-order varieties which increase both in accuracy and cost. These varieties are characterized by how many energetically-similar groups of lattice sites they represent, and are shown in Figure 2.18.

For simulations with large coarsened domains, the largest energetically-similar group of lattice sites to capture are the sites within the large elements, all of which see the same neighborhood (because of affine deformation within elements). In the **zeroth-order** optimal summation rule, sampling atoms are only located in element barycenters and the Cauchy-Born rule is applied to the neighborhood of the sampling atoms. This summation rule correctly represents the energy of the lattice sites that are more than the potential's cutoff radius away from the sides of the elements, but incorrectly represents any lattice site near element sides or edges. This rule represents the traditional local QC formulation which does not have sampling atoms on repatoms and is difficult to reduce down to atomistics.

The zeroth-order optimal summation rule does not account for either the repatoms (which see the deformation of many elements) or the element faces (edges in 2D) which see the deformation of two elements. Of these two groups, the element faces represent a larger group of energetically-similar lattice sites than the repatoms and it would seem natural to improve the accuracy of the summation rule by next including this group. However, the inability of the zeroth-order summation rule to reduce down to atomistics is too constraining and in the **first-order** optimal summation rule we add sampling atoms on repatom positions in order to easily transition to atomistics. The lattice sites around repatoms all experience different neighborhoods, but the energy of the repatom in the center is taken as an average or representative energy for all of the lattice sites within a given radius, the `nodeRepresentationRadius` (abbreviated in the following as r_n). This radius is a parameter of the summation rule and controls how much the sampling atoms on the repatoms contribute to the total energy.

In the **second-order** optimal summation rule, sampling atoms are added on element faces (in 3D) or element edges (in 2D), where in large elements each sampling atom would experience the deformation of two elements. This accounts for the next-largest group of energetically-similar lattice sites after the group in the bulk. The neighborhood of the sampling atom is chosen to be Cauchy-Born with respect to both elements. That is, the half of the neighborhood that lies on either side of the separating plane has the deformation of that element applied, whether or not those neighborhood sites actually fall outside of that particular element.

4.5.1 First-order weights in 2D

So far, we have only discussed the locations of optimal sampling atoms but not their weights. The weight of each sampling atom represents the number of lattice sites represented by the sampling atom's energy. If we consider the case of using the first-order rule with large elements such that no two repatoms are within $2r_n$ of each other (see Figure 2.18c), then the weight of all sampling atoms would be simply the product of the area of the circle with radius r_n (or sphere in 3D) and the density of lattice sites per unit volume. Weights of the barycenter sampling atoms can be determined by using the angles subtended by the sides of the triangles (in 2D) to determine how much of each element has already been accounted for by the sampling atoms on the repatoms; the weights of the barycenter sampling atoms are simply the volume left in each element after removing the portions that were accounted for by the sampling atoms on the repatoms. However, domains without atomistic regions are not the target of this work and some way of calculating weights must be devised that approximates this type of exact calculation even when repatoms are within $2r_n$ of each other.

The weight calculation algorithm must have the following characteristics:

- assign weights of 1 to sampling atoms in atomistic regions.
- assign constant weights of $\rho * V_d(r_n)$ (where ρ is the density of sites per unit volume and $V_d(r_n)$ is the volume of the ball of radius r_n in dimension d) to sampling atoms on repatoms that are farther than $2r_n$ from all other sampling atoms on repatoms.
- assign weights to the barycenter sampling atoms that represent the “remaining” volume of each element.
- transition seamlessly between the atomistic and coarse regions because the formulation does not distinguish between the region types.

Listing 4.6: Determining sampling atom weights by polytope trimming

```

for each element
  for each sampling atom incident on the element
    Polytope  $\mathbf{p}$  = element
    trimmingPlanes = define a set of planes for this sampling atom type
    for each plane in trimmingPlanes
       $\mathbf{p}$  = trim  $\mathbf{p}$  with plane
    volumeContribution = compute volume of  $\mathbf{p}$ 
    samplingAtomWeights[samplingAtom] += volumeContribution

```

Every sampling atom has weight contributions from several elements. The strategy taken in the current realization of the algorithm is to trim each element to determine the region of that element that contributes to each sampling atom, then take the volume of the trimmed region and add that volume to the sampling atom's weight. Each element starts as a polytope and is trimmed by a set of trimming planes which are specific to the element and sampling atom type. This process is enumerated in pseudocode in Listing 4.6.

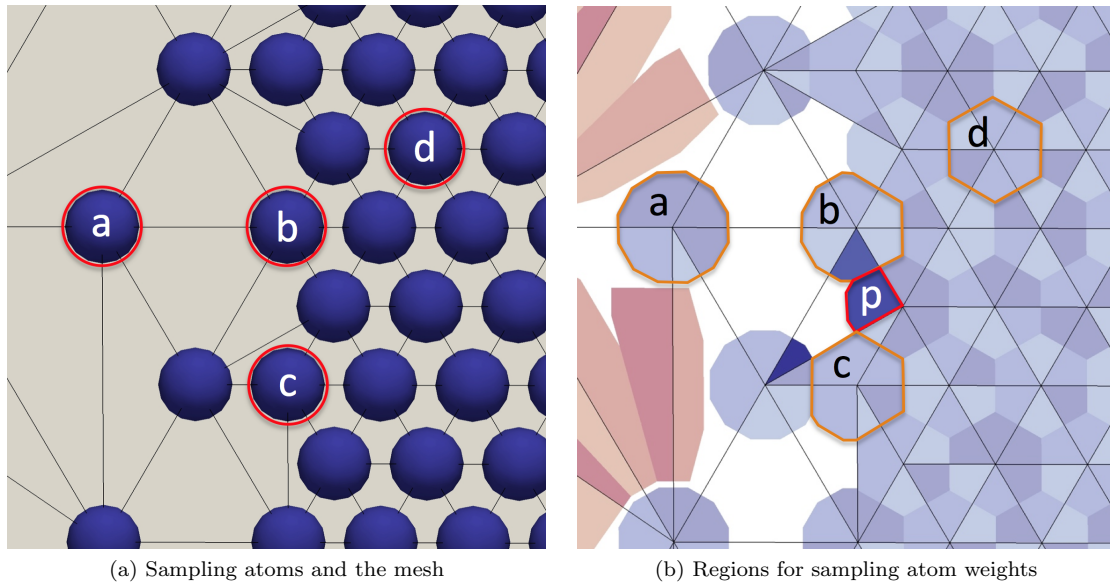


Figure 4.8: Illustration of regions used to compute optimal summation rule weights.

Consider the illustration shown in Figure 4.8. The two panels show the same physical space; the left panel shows the sampling atoms and the mesh and the right panel shows the regions that are used for sampling atom weight calculation. On the right side of each panel is atomistic resolution, and the domain is coarsened on the left side. For now, ignore the red and orange regions. Sampling

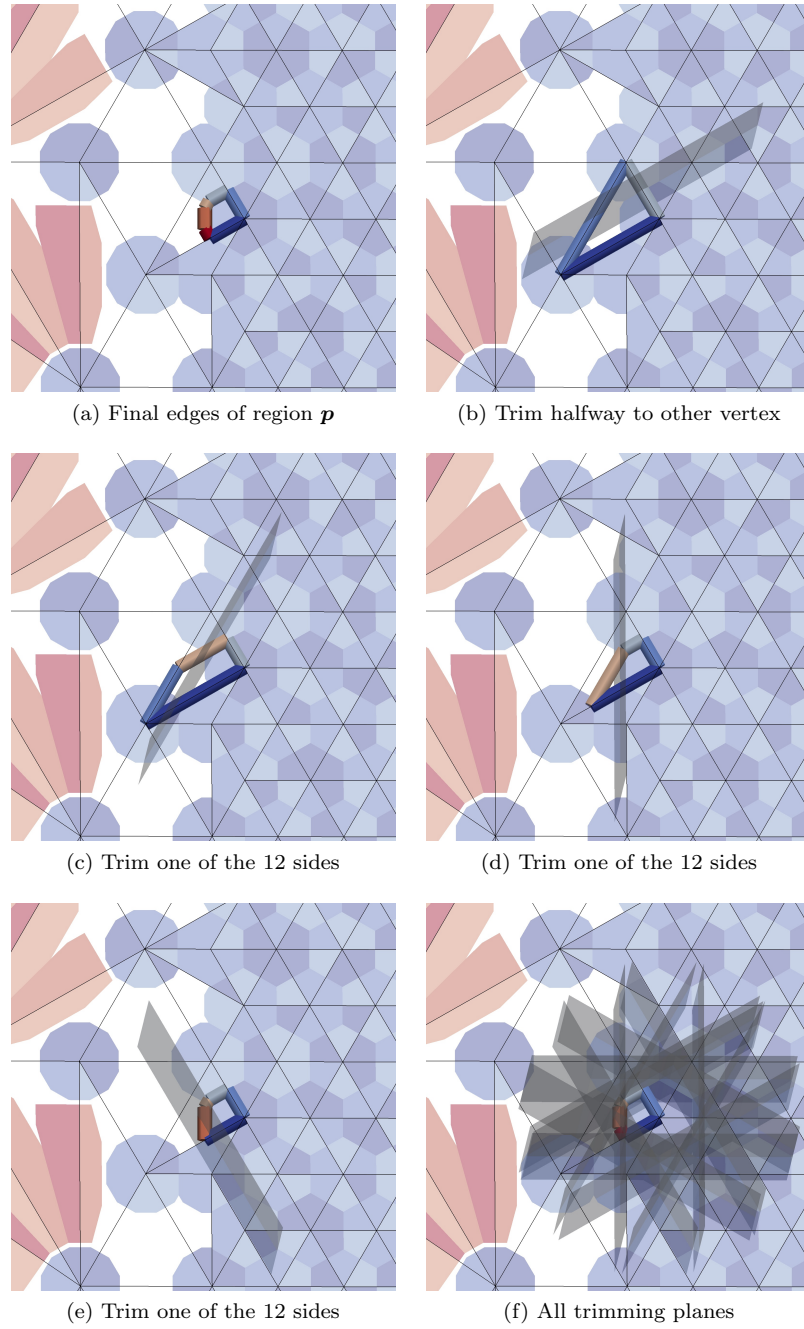


Figure 4.9: Trimming required to obtain region p . A movie is available [here](#).

atom a is more than $2r_n$ away from other sampling atoms, so its weight region does not overlap with other sampling atoms. Sampling atom d is within the atomistic portion of the domain and its weight region overlaps with other sampling atoms on all sides. Sampling atoms b and c are transition sampling atoms that overlap on some sides.

It should be noted that the regions obtained for sampling atoms a and d are extremely similar

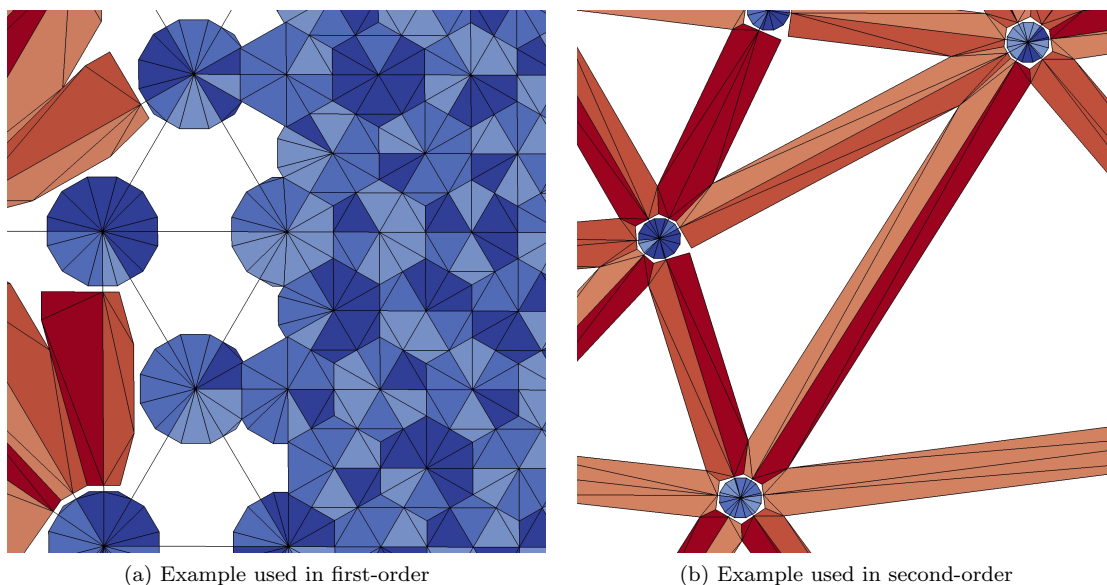


Figure 4.10: Sampling atom regions are meshed to find their volume.

and do not justify the special trimming logic simply to capture their subtle difference. However, the images produced for this example used a `nodeRepresentationRadius` that was smaller than is normally used, for the purposes of producing these images at a scale where coarsened, transition, and atomistic sampling atoms could be seen in one frame.

The balls around sampling atoms on repatoms are approximated as n -sided regular polygons (12-sided polygons are used in practice). The region associated with a given sampling atom is made up of pieces (contributions) from several elements, one of which is highlighted red and labeled \mathbf{p} . Once each piece (such as \mathbf{p}) is obtained, its volume is measured and added to the weight of the corresponding sampling atom. Pieces such as \mathbf{p} are obtained by starting with the element as a polytope. Then, the 12 trimming planes (which are the only necessary trimming if the sampling atom is isolated enough) are applied; any volume outside of those trimming planes is discarded. However, this is not sufficient to prevent overlap, and planes halfway between each pair of vertices in the element must be added. The process of obtaining \mathbf{p} is shown in Figure 4.9 and can be seen as a movie [here](#).

4.5.2 Second-order weights in 2D

The second-order optimal summation rule includes sampling atoms on the edges of elements. In order to calculate the contributions from each element to these side sampling atoms, the same strategy of trimming planes is employed. The trimming planes include the angle bisectors, planes to remove overlap from nodes, and the plane offset from the face. The process for obtaining an illustrative example region is shown in Figure 4.11. Once each region is found, its volume is added

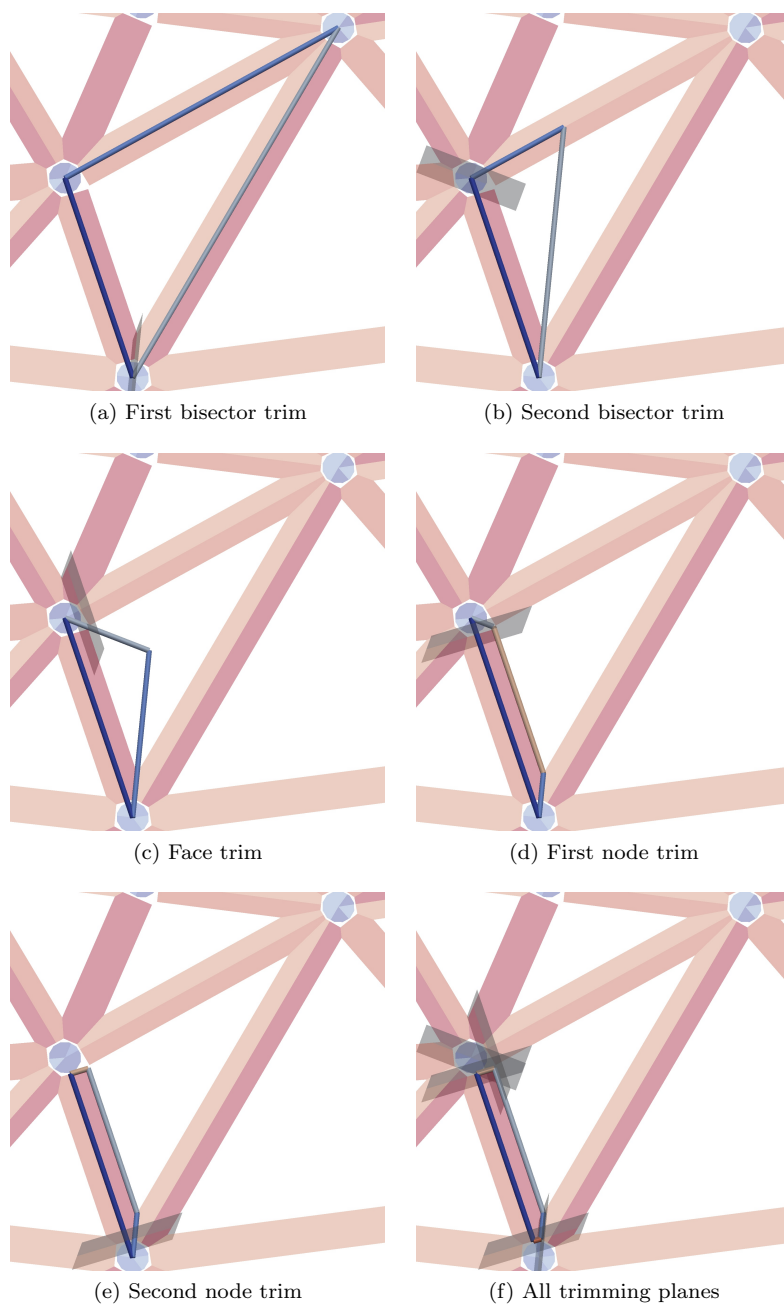


Figure 4.11: Trimming required to obtain the region for a side sampling atom. A movie is available [here](#).

to the corresponding sampling atom. To find the volume, the regions are meshed, examples of which are shown in Figure 4.10.

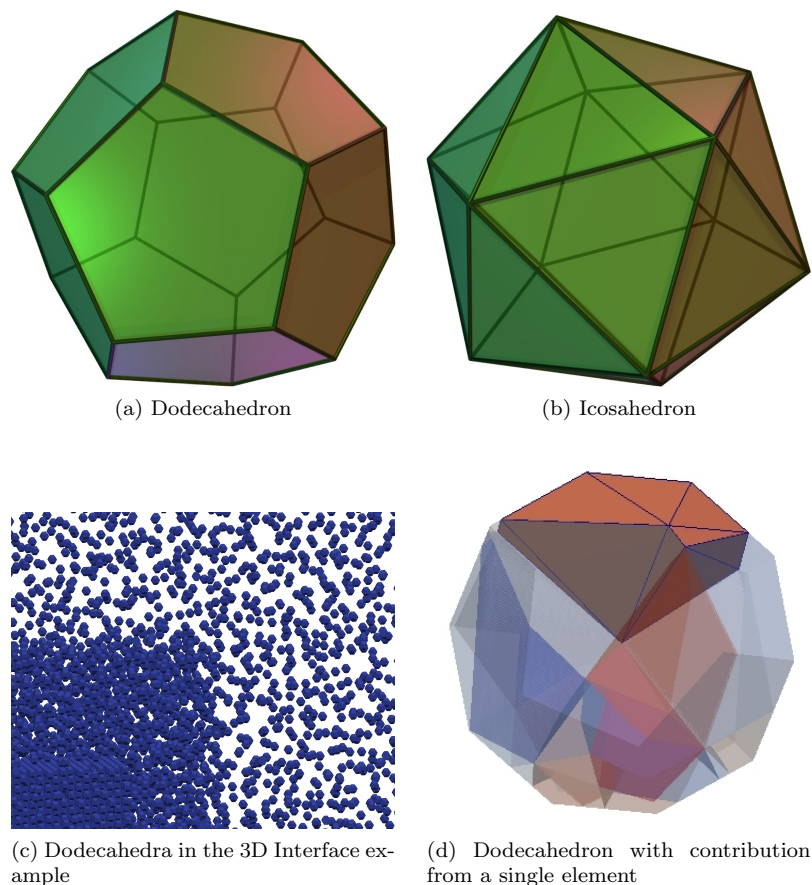


Figure 4.12: Dodecahedron approximation for 3D optimal summation weights.

4.5.3 Optimal summation rule weights in 3D

First-order In order to calculate the weights of sampling atoms on repatoms with trimming planes in 3D, we need an approximation to a sphere that is computable by a collection of trimming planes. The approximation chosen is the regular dodecahedron (see Figure 4.12a). The trimming planes required to produce a dodecahedron are found using its geometric dual, the icosahedron (see Figure 4.12b). Each plane is defined by an icosahedron vertex and the vector from the center of the icosahedron to the vertex. The process of trimming an element to determine the contribution to a sampling atom on a repatom is shown in Figure 4.13.

Second-order The second-order optimal summation rule includes sampling atoms on the faces of elements, and the same strategy of trimming planes is employed. For side sampling atoms, planes are added bisecting dihedral angles, offset from the face, and to separate the region from the incident nodes. Figure 4.14 illustrates example contributions from elements to side sampling atoms. It is difficult to demonstrate all of the regions that contribute to the sampling atoms with pictures, but a movie showing a cut-through of the 3D analog of Figure 2.19 is found [here](#).

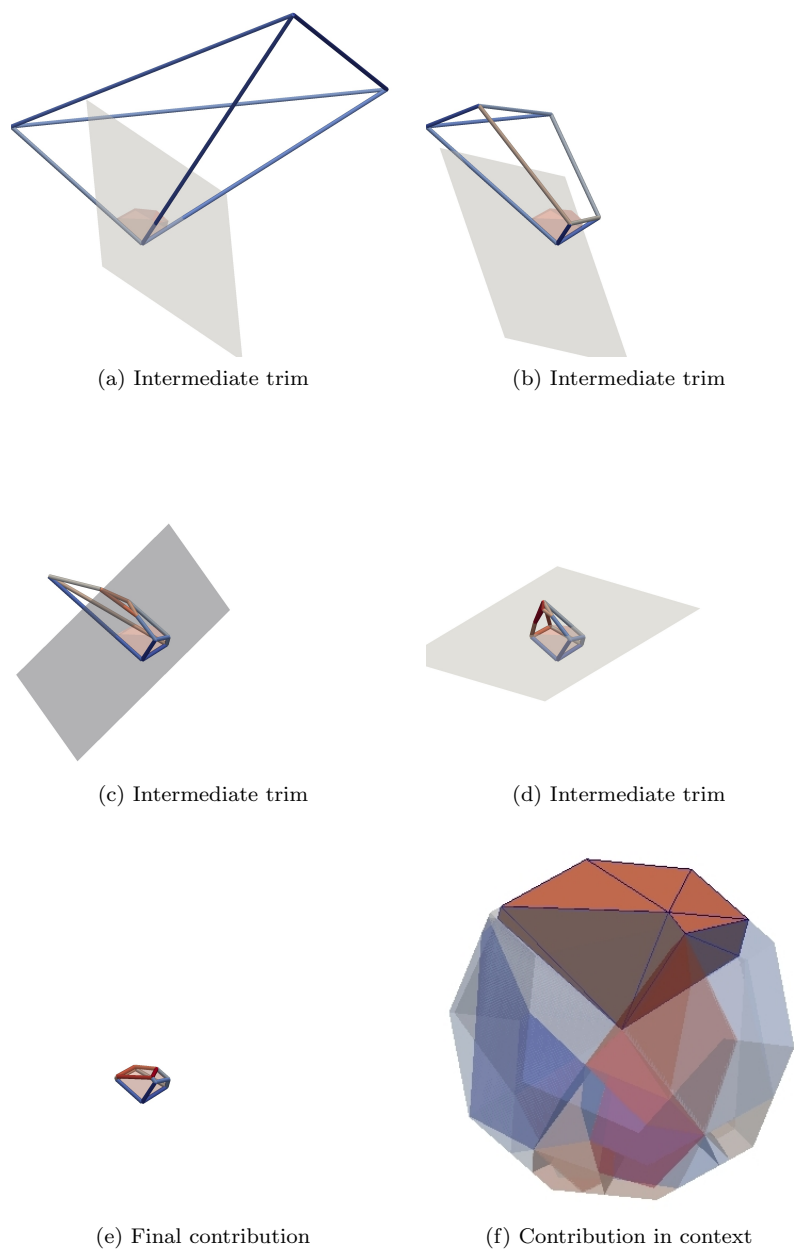


Figure 4.13: Trimming required to obtain a contribution to a sampling atom on a representative atom in 3D. A movie is available [here](#).

4.6 High deformation adaptive neighborhoods

Whenever energetic quantities are calculated at sampling atoms, those sampling atoms must have as their neighbors all lattice sites that are within the potential's cutoff radius, or the energetic quantity will be incorrect. In molecular dynamics, distance searches are simply performed within the pool of all atoms in the simulation to find all atoms within the cutoff radius. In the QC method, the required neighbors of any given sampling atom are not in any given pool and cannot be found using

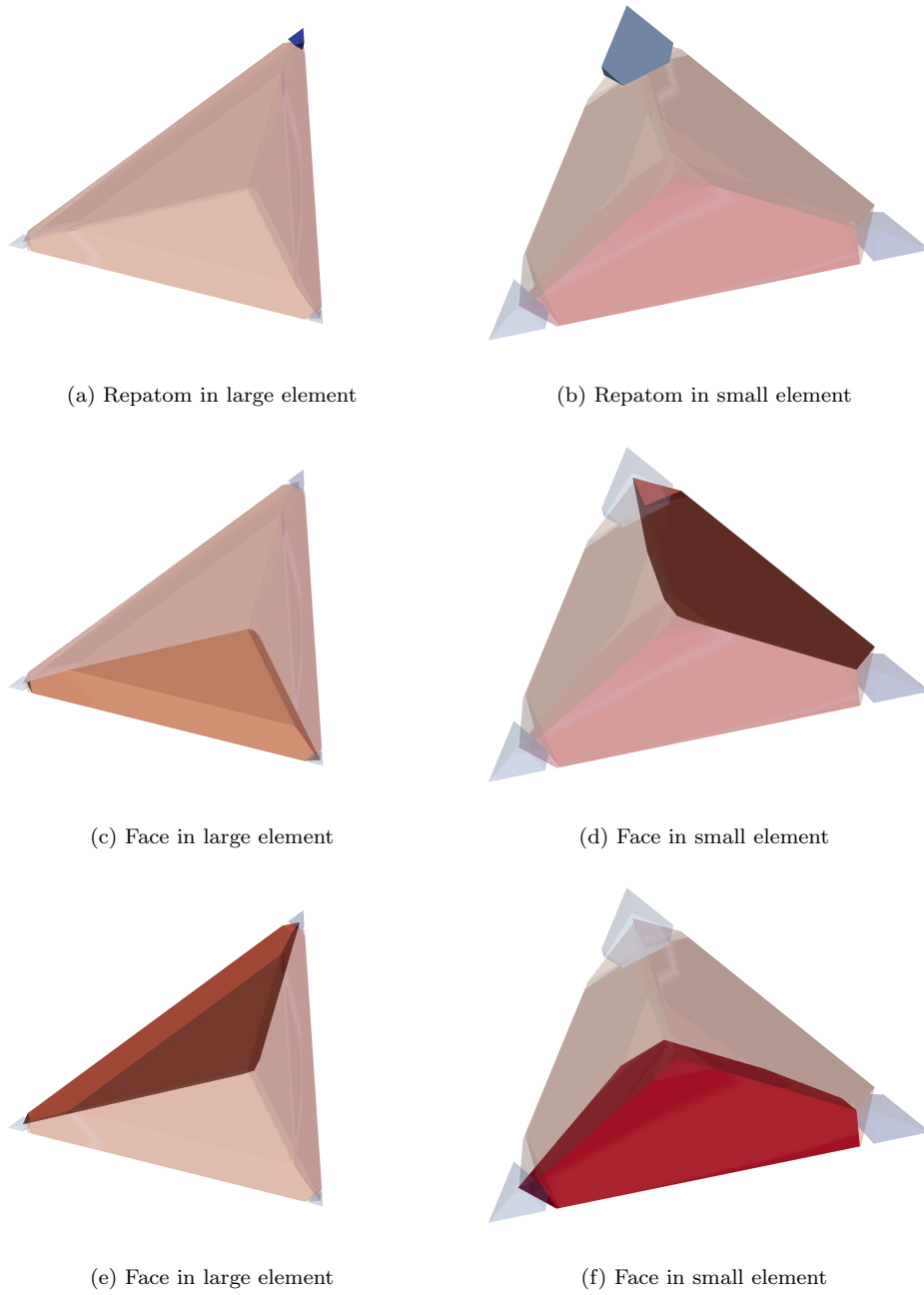


Figure 4.14: Example second-order weights for large and small elements, relative to the potential's interaction distance.

distance searches. Instead, all necessary neighbors are generated in the undeformed configuration with repatom dependencies and their positions are updated whenever repatoms move. It is vital that every sampling atom has its complete neighborhood in the deformed configuration, so the selection and population of neighbors in the undeformed configuration is a crucial step and will be explained in this section.

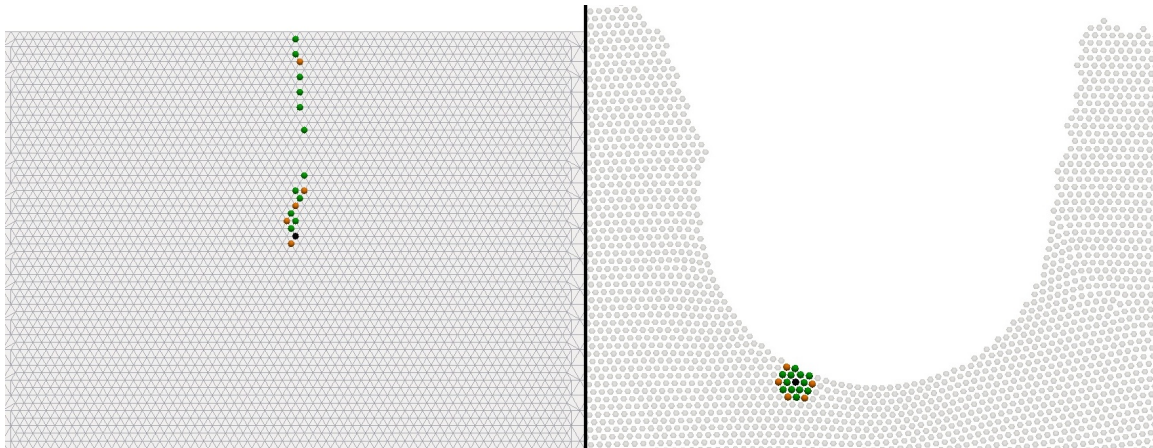


Figure 4.15: Example of the neighborhood of a sampling atom in the undeformed (left) and deformed (right) configurations during a nanoindentation simulation. For an explanation of the color coding and scenario, see Appendix A. A movie is available [here](#).

In this QC formulation, each neighbor is a set of “repatom dependencies”, which are pairs of repatom indices and shape function values. Neighbors that lie on repatoms have only one pair, neighbors which lie on edges have two pairs, and neighbors which lie in the interior of elements have $\text{SpatialDimension}+1$ pairs. Whenever sampling atoms are added or changed, their neighborhoods are generated and stored. After every time that repatoms move, the positions of sampling atom neighbors are computed using the repatom dependencies: each position is the sum of the shape function values times the updated position of the corresponding repatom.

For many common EAM potentials, this means that each sampling atom will need to generate and keep track of around 20 neighbors. If the simulation is purely elastic, the 20 neighbors from the beginning of the simulation will most likely be sufficient for the entire simulation. However, if a simulation experiences any plastic deformation, the sampling atoms with highly deforming neighborhoods will have to update their neighborhoods with new neighbors.

In order for a sampling atom to maintain a correct neighborhood in the deformed configuration, it must generate more candidate neighbors from farther away in the undeformed configuration. It is possible for sampling atoms to keep all of these candidate neighbors as their neighborhoods. However, this can lead to extremely expensive simulations, as sampling atoms may need to maintain tens of thousands of neighbors for highly-deforming 3D simulations. Instead, the candidate neighbors are mapped to the deformed configuration, and only those candidates which lie within a certain deformed cutoff radius are kept. An example of the neighborhood of a highly-deforming sampling atom in the undeformed and deformed configurations is shown in Figure 4.15.

Knowing when a sampling atom needs to rebuild its neighborhood is difficult; this must be determined from the only information it has, which is the positions of its neighbors. In the current realization of the algorithm, neighborhoods are regenerated if the relative position from a sampling

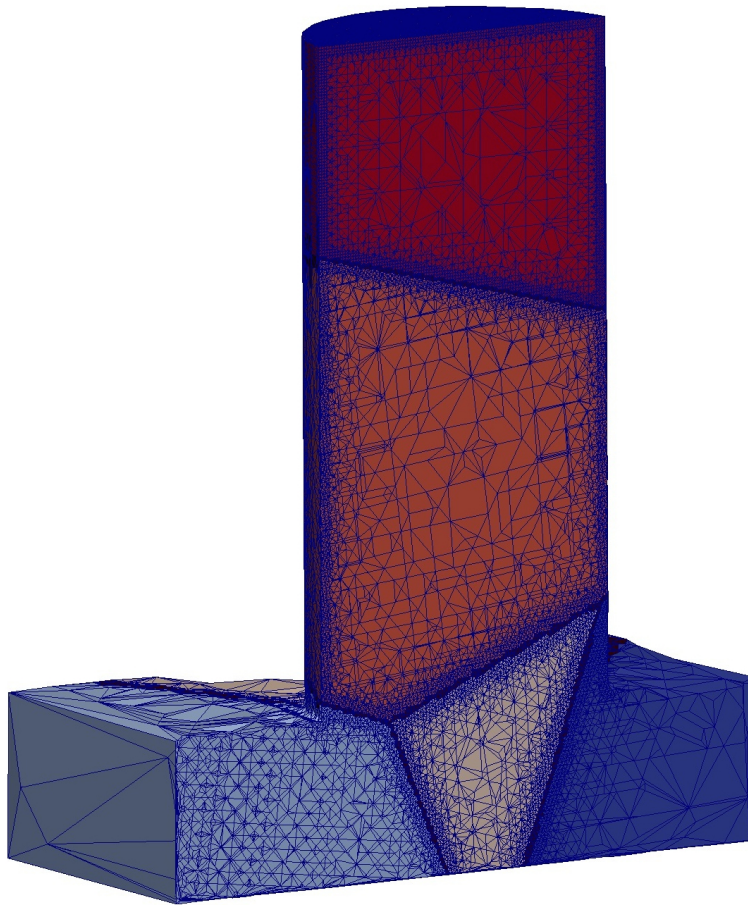


Figure 4.16: An example polycrystal domain. A nanopillar of radius 10nm and height 40nm is formed by 10 grains. A movie of the domain is available [here](#).

atom to any of its neighbors has changed sufficiently. The algorithm is successful at maintaining correct neighborhoods in the presence of even extreme deformation, as can be see in [this](#) movie. In Appendix A, we discuss in full detail the algorithm implemented to determine when to rebuild neighborhoods.

4.7 Polycrystals

Many systems of engineering or scientific interest are composed of crystals of separated by grain boundaries. In order to support such polycrystal systems, every element in the simulation has an associated orientation. Both neighborhood generation (described in Appendix A) and remeshing (described in Section 4.8) generate candidate lattice sites using a vertex of an element and the element's Bravais lattice vectors. Therefore, the vertices of any given element as well as any implied lattice sites contained within the element must form a Bravais lattice with the corresponding orientation.

The vertices of elements which lie on grain boundaries do not form a Bravais lattice, and no algorithm should attempt to generate lattice sites within these elements. To prevent this from happening, elements which should not be used to generate lattice sites are marked with invalid orientations. Figure 4.16 shows a cut-through of a polycrystal nanopillar, and a corresponding movie can be found [here](#). In these visualizations, each grain is colored differently and elements with invalid orientations are colored dark red. A thorough study of polycrystal scenarios with the QC formulation of this thesis is being performed by Ishan Tembhekar in the Kochmann research group.

4.8 Remeshing

One of the most useful aspects of the QC method is the ability to model microstructure and other nanoscale features in simulations of large physical spaces. In order to model a feature such as a dislocation, atomistic resolution is necessary around the dislocation (i.e., all lattice sites must be repatoms). However, dislocations move under applied stress and the coarsened regions of a QC mesh are unable to kinematically represent dislocations. If dislocations move towards a coarsened region, atomistic resolution must be introduced in its path or the artificial and nonphysical trapping of the dislocation corrupts the results of the simulation. A key component of a QC formulation designed to model plastic deformation is therefore the ability to refine the mesh in order to provide atomistic resolution around any microstructure.

In this QC formulation, a simulation is divided into a set of “loadsteps”, or configurations which represent incremental changes in external forces and boundary conditions. At each loadstep, the solver either finds the solution to $\mathbf{F} = 0$ (for quasistatic simulations) or $\mathbf{F} = m\mathbf{a}$ (for dynamic simulations). In quasistatic simulations, each loadstep represents a full equilibration. If a dislocation exists in the simulation and the applied loads change, full equilibration may cause the dislocation to propagate through the entire simulation domain. For example, if the nanofilm simulations of Section 3.5.3 are performed quasistatically, it is common that the entire sample will behave elastically until it cleaves in two pieces over the course of a single loadstep. In order to capture the growth and shape of microstructure, simulations can be performed using implicit dynamics, which limits the speed at which microstructure can grow but also restricts simulations to high strain rates.

For both quasistatic and dynamic simulations in which microstructure is present and can grow or move, there must either be sufficient initial atomistic resolution (if the simulation is performed without remeshing) or remeshing must be performed within the iterations of the solver. Providing sufficient initial atomistic resolution is expensive and requires a priori knowledge of microstructure shape and propagation direction; adaptive remeshing during the course of a loadstep is the preferred method of allowing for natural microstructure evolution.

Two main strategies have been pursued to perform remeshing within the current realization:

global and local remeshing. The metric used in both strategies for determining which elements are to be refined is based on the criterion proposed in [Knap and Ortiz, 2001]: for each element in the mesh the second invariant of the Lagrangian strain tensor is computed and compared to a tolerance level. However, there is room for improvement here and criteria involving sampling atom energies or centrosymmetry values may be better indicators.

Listing 4.7: Global remeshing process

```

for each element we'd like to refine
  for each edge that is similar in length to this element's longest edge
    find midpoint of edge
    find lattice site nearest to midpoint that does not already contain a vertex
    add lattice site to undeformed reptoms

interpolate deformed positions of new reptoms

collect all undeformed reptoms onto a single instance of the program (coordinator)
if (this instance is the coordinator)
  create mesh of all undeformed reptoms using external mesher
  partition mesh into pieces, one for each instance of the program
  send pieces to instances
else
  receive this instance's mesh from the coordinator

collect all previous sampling atoms onto all instances of the program
populate sampling atoms for this instance's mesh
for every new sampling atom
  if there used to be an old sampling atom in the same location
    new sampling atom's maxNeighborDistance = old's maxNeighborDistance
regenerate sampling atom neighborhoods

```

4.8.1 Global remeshing

The product of any remeshing algorithm is a modified mesh containing new vertices that have been chosen to provide sufficient resolution in required areas. This modified mesh can either be created entirely anew by an external mesher, or it can be produced by performing local modifications to the existing mesh. In this section, we consider the strategy of creating an entirely new mesh each time that new vertices are added.

The global remeshing algorithm is similar to the process of parallel initialization of a new simulation: in both situations a set of repatoms is produced (either by the scenario for parallel initialization or by adding to the nodes of a mesh in the case of global remeshing) and all of the associated data structures must be built. In particular, sampling atoms must be re-generated with respect to the new mesh, which may be significantly different than the old mesh. Because sampling atoms have an associated parameter (the `maxNeighborDistance` described in Appendix A.3.4) which stores how highly deformed the sampling atom's neighborhood is, this information must be transferred from the old set of sampling atoms to the new set of sampling atoms. Otherwise, new sampling atoms placed in the same locations as old sampling atoms will not populate their full neighborhoods. The process for performing global remeshing is enumerated in Listing 4.7.

The process of remeshing may happen as often as every few iterations of the solver, and global remeshing is an expensive process. When the simulation is being run in parallel, global remeshing requires substantial amounts of communication and coordination which eclipse the cost of solver

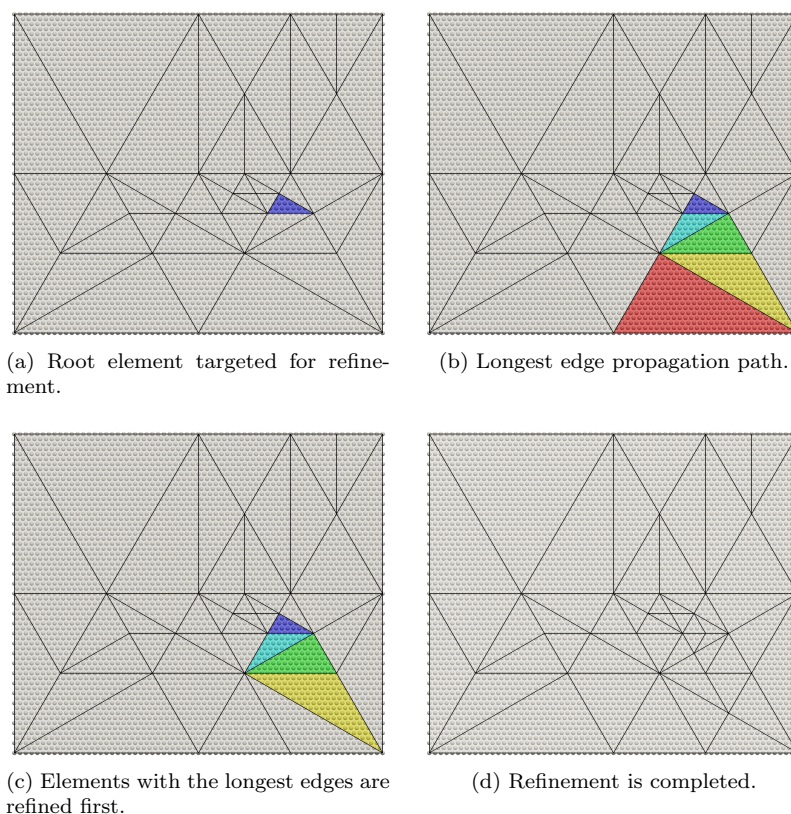


Figure 4.17: Illustration of the longest edge propagation path refinement algorithm. In order to bisect the root element (a), the Lepp is formed (b), refined in order of longest edges (c) until the root element is refined (d). Spheres denote positions of sites in the underlying lattice; new mesh vertices may only be placed on these lattice sites. A movie of this process is available [here](#).

iterations. Therefore, in practice we only use global remeshing in small serial simulations; global remeshing is prohibitively expensive for large serial simulations or any size of parallel simulation.

4.8.2 Local remeshing

In global remeshing, it is egregiously wasteful to create an entirely new mesh when only a few new repatoms need to be inserted. Instead, one would rather modify the existing mesh locally, a process which may not even incur communication and coordination. This is the idea and driving motivation behind local remeshing.

There is a wealth of literature on the subject of mesh refinement for finite element simulations in which new vertices may be inserted anywhere in space to improve the mesh (see [Bern and Plassmann, 2000] and [Jones and Plassmann] among many others). Longest edge bisection, in which simplices are bisected by the median of the longest edge, is a common strategy used in many of these algorithms. The changes required for an edge bisection are localized: only the elements within the element star of the edge are changed. However, simply bisecting the one element which is selected for refinement leads to meshes of bad quality, and more sophisticated techniques are needed to maintain quality triangulations.

Listing 4.8: Discrete Lepp-bisection algorithm

```

sort list of elements to refine by their longest edge
for each rootElement from the one with the longest edge to the shortest
  elementStack = []
  elementStack.append(rootElement)
  while (elementStack is not empty)
    element = elementStack.top()
    longestEdge = find longest edge of element
    star = find element star of longestEdge
    if longestEdge is the longest edge of all elements in star
      newVertex = lattice site on the interior of the star which is
        closest to the midpoint of longestEdge and which does not
        already have a vertex there.
      bisect longestEdge with newVertex
      elementStack.pop()
    else
      push all elements in star onto elementStack

```

A common mesh refinement technique is the longest edge propagation path (Lepp) bisection class of algorithms [Rivara, 2009]. In Lepp algorithms, the elements in a star around edge e are

not split until e is the longest edge in all of the elements of the star. That is, in order to refine a particular edge, the refinement propagates along the path of longest edges. This process is illustrated in Figure 4.17.

Lepp refinement does not require an entire re-triangulation of the vertices and is a candidate for the local remeshing purpose here. However, the longest edge propagation path can extend arbitrarily far through the domain. It is precisely this behavior which provides Lepp algorithms with one of its key characteristics: bounds on the quality of elements produced by the refinement. It is possible to terminate the Lepp process after recursing to a certain level, but this runs the risk of producing elements of large condition number.

A complication in using any mesh refinement algorithm is the discrete nature of a QC mesh: mesh vertices may only be placed on valid lattice sites. One strategy is to give the deformed QC mesh to a traditional mesh refinement tool, which will insert points, perform bisections, do edge

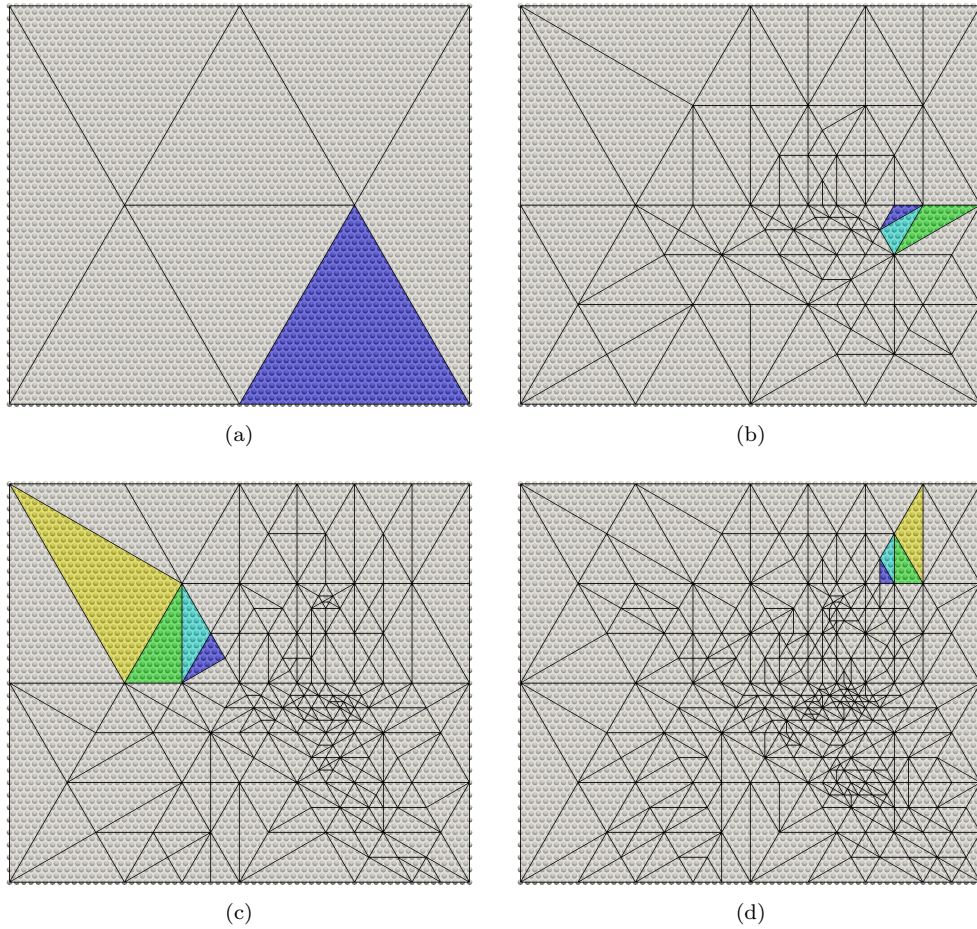


Figure 4.18: 2D Lepp refinement example. Each frame is taken from illustrative steps in the refinement; colors indicate depth in the propagation path for a particular path. Random elements are being chosen for refinement as a test to produce a good atomistic mesh. A movie is available [here](#).

flips, and apply other techniques. The returned mesh will have points which are not on lattice sites, but one could then snap these points to their nearest atomic sites.

This strategy of using external refinement software and snapping points to lattice sites is not used in the current realization of the formulation. First, the mesh would be refined in the deformed configuration, but in this QC formulation the mesh is used for interpolation in the undeformed configuration. New points created by the refinement tool would need to be mapped back to the undeformed configuration and snapped to atomic sites, and the connectivity created by the tool in the deformed configuration may be invalid for interpolating in the undeformed configuration. Secondly, the mesh refinement tool may insert more points in a given region than there are valid atomic sites. One could duplicate points that are snapped to the same atomic site, but connectivities would have to be modified. Lastly, snapping points to nearest atomic sites may invert simplices.

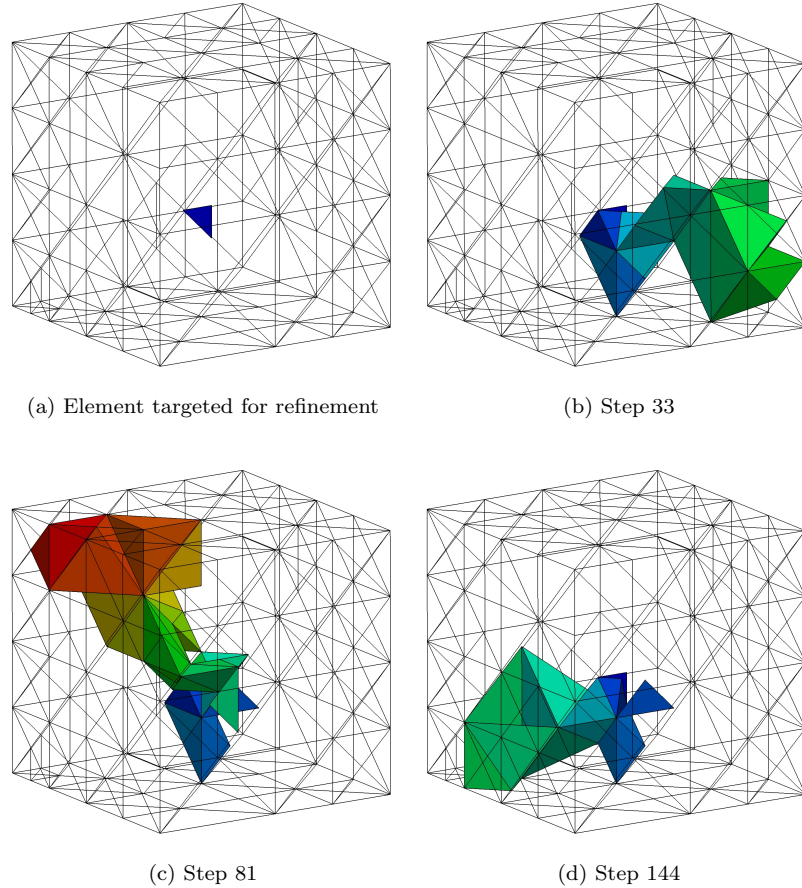


Figure 4.19: 3D Lepp refinement example of a single element. Each frame is taken from illustrative steps in the refinement of the one target element; colors indicate depth in the propagation path for a particular path. The total number of elements bisected was more than 300 and included elements on all 6 faces of the domain's boundary. A movie is available [here](#).

For all of these reasons, we are not able to use external refinement software and must perform the refinement ourselves.

The discrete Lepp-bisection algorithm we implemented is shown in Listing 4.8, from which some details have been omitted for concision. An example of the performance of the discrete Lepp-bisection in 2D is shown in Figure 4.18 and can be seen as a movie [here](#). In this example, an initially-coarse mesh is refined down to atomistics by choosing random elements. The longest propagation path in the refinement can be seen [here](#). A 3D example can be seen [here](#), the longest propagation path of which is shown in Figure 4.19 and can be seen as a movie [here](#).

Ultimately, we determined that Lepp-style algorithms are not suitable for local remeshing of this QC formulation and instead focus on a more localized refinement technique to be described later. The reason for not choosing a Lepp-style algorithm can be seen in Figure 4.19: the longest edge propagation path used to refine a single element in a domain may extend across the entire simulation.

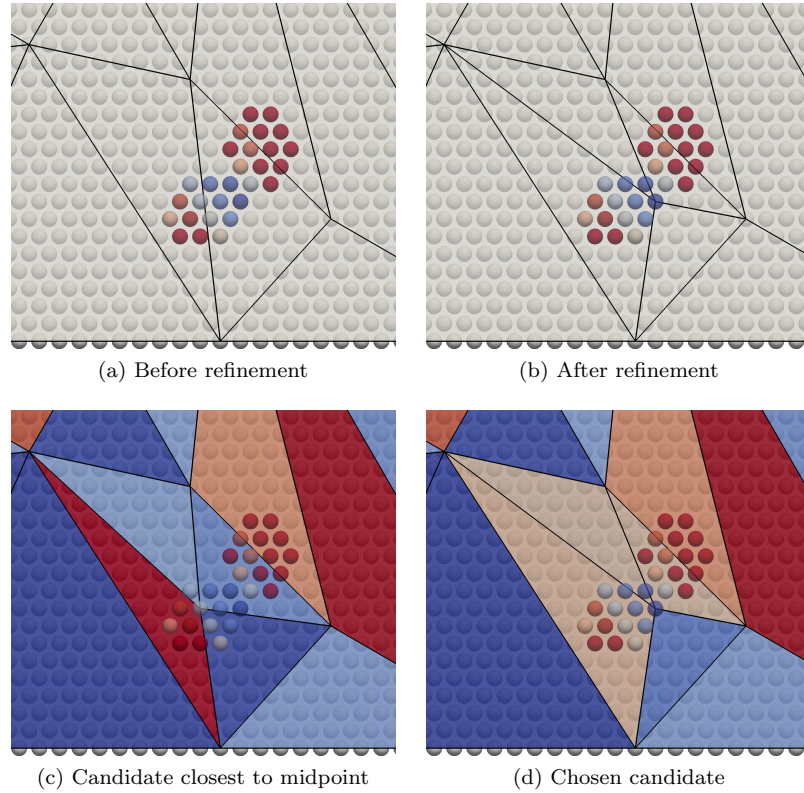


Figure 4.20: Existing local remeshing algorithm. The underlying lattice is shown as transparent white spheres, the candidate new vertices are spheres colored by their quality, and elements are colored by their condition number. The refinement performed (b) is far from the edge midpoint, which would be used by traditional methods. The algorithm chooses the new vertex because it results in the minimum max condition number within the new simplices. That is, the max condition number in (d) is less than the max condition number in (c). A movie of all of the candidates and their respective simplex qualities is available [here](#).

In the particular example from the figure, in order to refine the one chosen root element, refinement had to be performed on elements that touch all six sides of the simulation domain's extrema, and more than 300 elements were bisected in total.

As described in Section 4.4, in the parallel formulation of this QC method every instance of the program only stores a subset of the domain. No single instance of the program stores the entire mesh because it cannot fit in the memory of a single computer. Many Lepp-bisection parallel algorithms ([Rivara et al., 2004], among others) depend on all instances of the program storing the entire mesh with a central program instance assigning or coordinating tasks to worker instances. Some parallel algorithms use distributed data structures but have communication patterns that may not be scalable to modern distributed-memory machines with hundreds of thousands of processors (see [Balman, 2006] and Cougny and Shephard [1999] among others). Active research is being performed by the Kochmann research group into local remeshing performed in the context of a mesh stored in distributed memory.

Listing 4.9: Local refinement algorithm

```

sort list of elements to refine by their longest edge
for each element from the one with the longest edge to the shortest
  newVertex = NULL
  minimumMaxConditionNumber = infinite
  for each side whose edge length is close to the longest edge length
    for each candidateNewVertex in a cluster around this edge's midpoint
      insert candidateNewVertex into the mesh
      maxConditionNumber = max(condition numbers of new elements)
      if maxConditionNumber < minimumMaxConditionNumber
        newVertex = candidateNewVertex
        minimumMaxConditionNumber = maxConditionNumber
  insert newVertex into mesh

```

We chose not to pursue the Lepp-style algorithm because a refinement of one element on one instance of the program could propagate through the mesh of an arbitrary number of other instances. The chosen technique for local remeshing was to make as localized a change as possible in order to minimize communication and coordination between instances of the program. When a new vertex is inserted, it only modifies its immediate neighborhood of elements (the star if the insertion causes an edge split, the two touching elements if the insertion is a face split, or solely the containing element itself if the insertion causes a simplex split). That is, we chose the local, greedy, hill-climbing algorithm of Listing 4.9: we consider all possible changes to the local region and choose the best one of those changes, regardless of the global situation.

This algorithm was designed to maximize locality, so that communication is reduced to a minimum. Because it is difficult to produce good meshes when changes are localized, the algorithm attempts to compensate for this anticipated shortcoming by performing the absolute best insertion it can find instead of simply using the lattice site nearest the midpoint. However, though the greedy hill-climbing algorithm is highly local and produces the best local modifications possible (see Figure 4.20, movie [here](#)), these local modifications produce meshes of exceptionally poor overall quality (see Figure 4.21b). In particular, the algorithm produces elements of high condition number which are difficult to refine to atomistics and block the path of dislocations.

4.8.3 Discussion

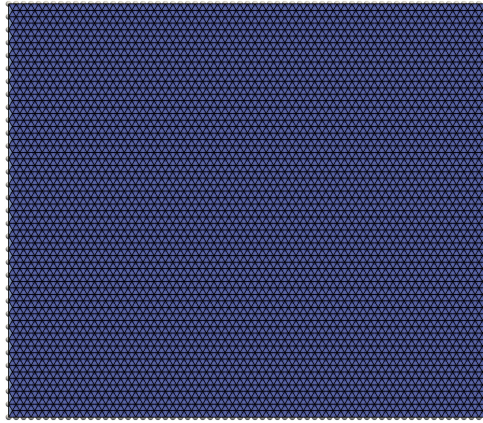
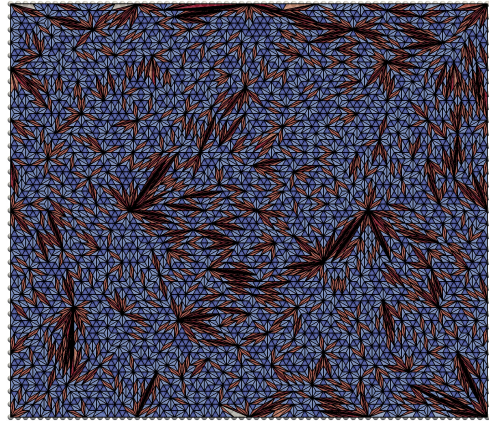
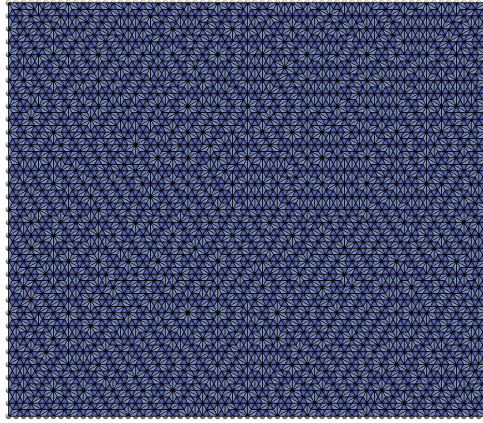
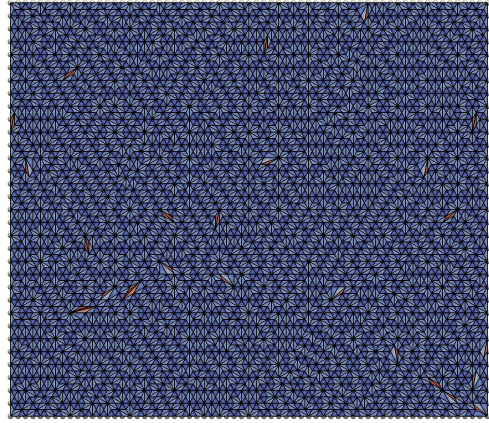
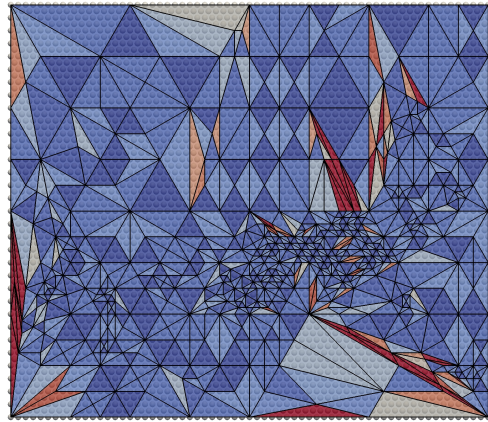
Figure 4.21 shows the results of a test of the refinement algorithms in 2D. Starting from a coarse mesh, remeshing is performed many times consecutively, each time on a random subset of 10% of the elements, with the goal of eventually reaching atomistic resolution everywhere. The behavior of the different meshers is evident: global remeshing’s meshes change drastically from one remeshing iteration to another, Lepp performs well when the terminal level is greater than 1, and local remeshing performs very poorly.

Remeshing is an essential component of the QC formulation, and it must be robust, fast, and scalable to thousands of processors. Most existing and popular methods of mesh refinement are not applicable to this QC formulation because mesh vertices must be placed on atomic sites and there is no complete copy of the mesh stored on any single processor. Global remeshing is too costly to perform in realistic simulations and serializes the parallel formulation of the method; a remeshing solution must only make local changes to the mesh that will not propagate drastically through the domain. In the current work, we have attempted the most local remeshing technique possible, but it produces meshes of prohibitively bad quality.

A performant, successful local remeshing scheme is the single most important improvement to make to the current QC implementation. Some possible directions to explore are reformulating existing parallel Lepp algorithms for fully distributed meshes, performing local constrained Delaunay triangulations of areas to be refined, or performing local advancing front triangulations of areas to be refined.

4.8.4 Examples

The examples in this section all use global remeshing because the implemented local remeshing technique generates meshes of prohibitively poor quality. Therefore, the simulations must be run in serial and contain small numbers of repatoms (i.e., fewer than 100,000).

(a) Global refinement (movie [here](#))(b) Local refinement (movie [here](#))(c) Lepp without terminal level (movie [here](#))(d) Lepp with terminal level 2 (movie [here](#))

(e) Lepp with terminal level 1

Figure 4.21: Comparison of refinement algorithms for the test case described in Section 4.8. Elements are colored by their condition number, where blue is a value of one and red represents a value of four or more. Lepp with terminal level 1 does not complete because it encounters a situation in which the triangle quality is so poor that bisection creates an interpenetrating mesh.

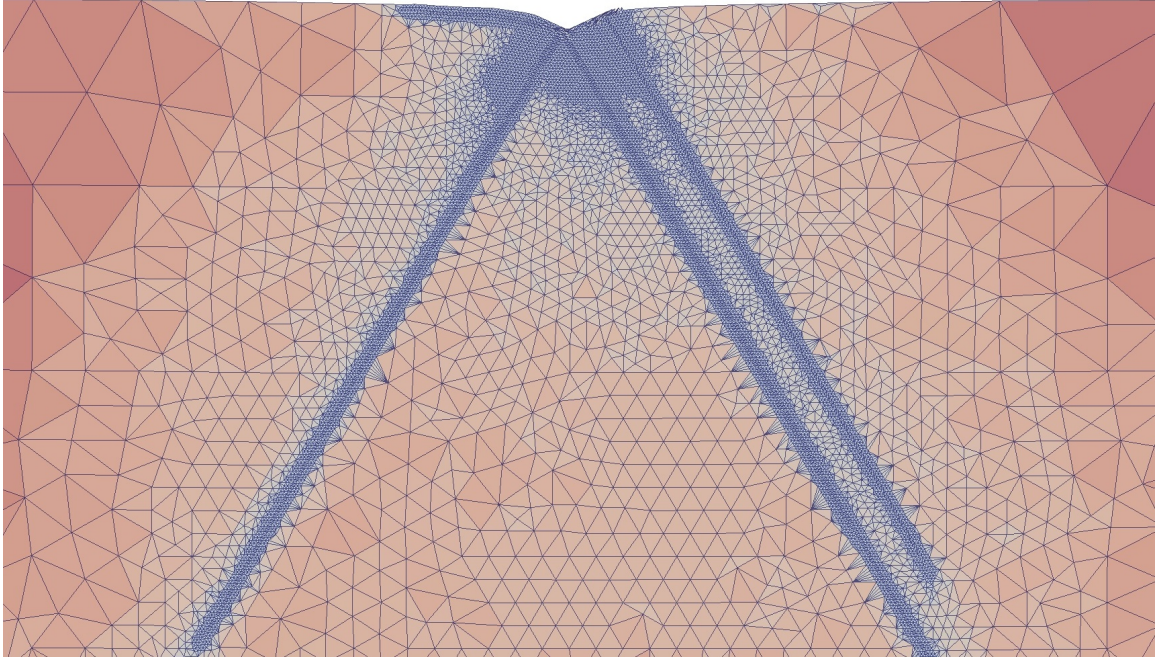


Figure 4.22: Remeshing performed during a quasistatic 2D nanoindentation example with a pyramidal indenter. The mesh is colored by the element size.

4.8.4.1 Nanoindentation

Nanoindentation is a canonical example to illustrate the behavior of adaptive model refinement schemes because the dislocation structures that form beneath nanoindenters quickly propagate through the domain (see Figure 4.22 for an example). In order to properly capture measurable quantities (e.g., force-displacement) without the corruption resulting from trapped dislocations, atomistic refinement is required in the path of the dislocations. For the domain to have this crucial atomistic resolution, either precise knowledge of dislocation paths can be used to appropriately place the atomistic resolution in the initial mesh, or the initial mesh must contain immense regions of atomistic resolution, or an appropriate remeshing scheme must be used.

The performance of automatic mesh refinement on 2D nanoindentation problems can be seen in Section 3.4.1, and an illustrative movie showing the behavior of remeshing, adaptive neighborhoods, and implicit dynamics can be seen [here](#). Global refinement performed during spherical 3D nanoindentation is analogous and is shown in Figure 4.23 and as a movie [here](#). In particular, each frame of that movie is another iteration in the refinement process and is not a loadstep of a simulation. In general, global remeshing is insufficient for 3D simulations and a performant, effective local remeshing technique is needed for large-scale 3D simulations.

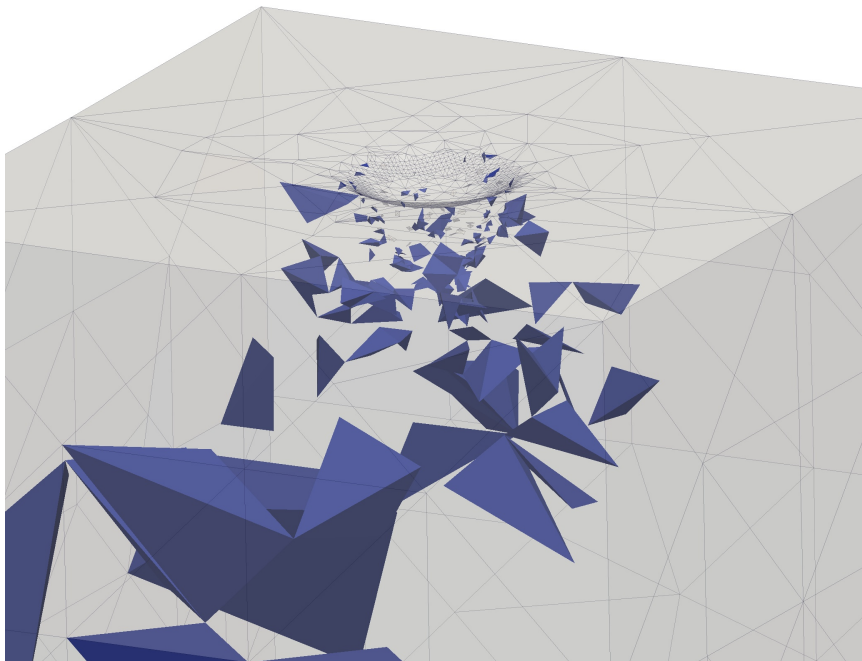


Figure 4.23: The elements selected for refinement during a 3D nanoindentation example. A movie is available [here](#).

4.8.4.2 Relaxation of a nano-cube

In order to test our mesh refinement capabilities, we will examine the refinement induced by surface relaxation. We quasistatically relax the initial configuration of a nano-cube of side length 11.52 nm whose edges align with the (100) directions of the Cu single crystal modeled by the extended Finnis-Sinclair potential of [Dai et al., 2006]. Surface relaxation leads to a deformation of the cube without any applied loads, and the refinement algorithm is applied to achieve higher accuracy near the surface of the cube. Figure 4.24a shows the resulting total number of sampling atoms in the QC representation of the cube compared to the total number of lattice sites in the cube. For large values of the refinement tolerance (effectively suppressing refinement overall), all QC models have the same underlying mesh and thus the number of sampling atoms. For example, a tolerance of 10 merely indicates the different numbers of sampling atoms resulting from the different summation rules. For small values of the refinement tolerance, all simulations approach full atomistic resolution.

The total energy of the relaxed cube contains essential contributions from atoms on and near the surfaces. Figure 4.24b shows the total error introduced by five different summation rules (the new second-order rule is included with two different choices of the effective interaction distance r). Of course, one cannot expect that large coarsening ratios (i.e., high refinement tolerances) reproduce the energy correctly, especially in such a small cube where the relaxation near edges and vertices deviates from the planar geometry studied in Section 3.5.1. Results show that for large element sizes

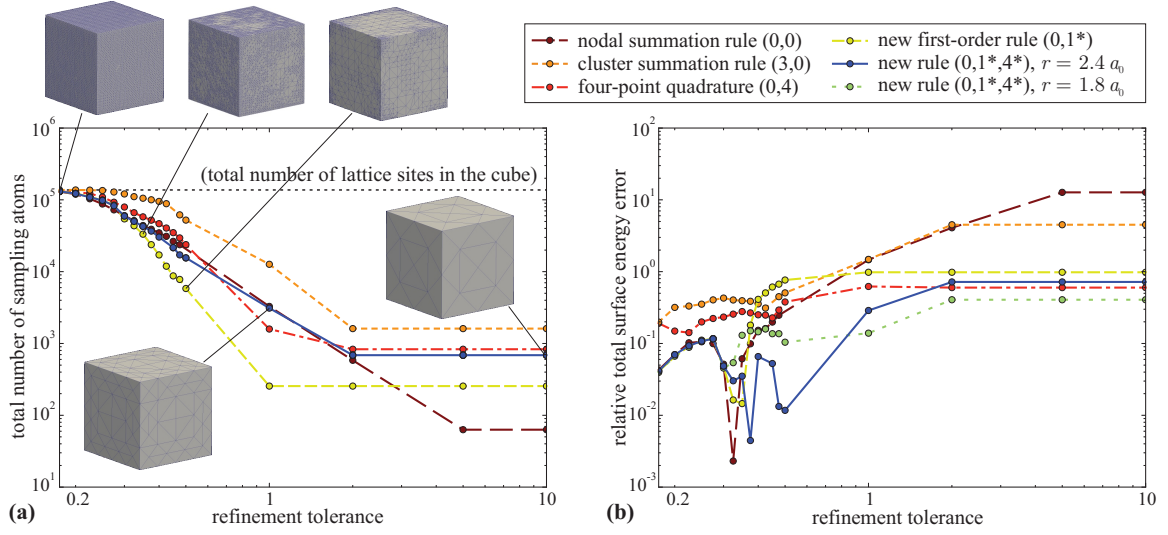


Figure 4.24: Relaxation of a free-standing nano-cube with adaptive mesh refinement: (a) total number of sampling atoms in the QC representation of the cube resulting from different values of the refinement tolerance (graphics show the cube after refinement for different refinement levels); (b) error of the approximated total surface energy of the cube vs. refinement tolerance for various summation rules.

(high refinement tolerance) the error produced by the new summation rules is orders of magnitude less than that of the nodal and cluster rules, and comparable to four-point quadrature. Even with refinement the new second-order rule overall performs best and offers a significant improvement.

Chapter 5

Discussion, Outlook, Conclusions

On the forefront of scientific research, new manufacturing technologies enable the production of materials with features and behavior previously-unimagined. The ability to create systems with abundant free surfaces (such as nanoporous metals, foams, and lattices) opens up new horizons in the study of physical, chemical, and mechanical behavior. There is a wealth of promising physical phenomena that leverage nanoscale features for tuning or customizing overall system response. Modeling the behavior of these new systems is difficult precisely because the overall systems response is a function of nanoscale behavior. In order to predict and analyze these effects, new tools are needed that are capable of tracking or accounting for atomic effects while not limiting the overall system size to unrealistically-small samples. Beyond the need to model new functional materials, a large number of small-scale deformation and failure mechanisms remain poorly-understood and require these new tools.

From its inception almost 20 years ago, the QC method has enabled the incorporation of atomistic phenomena into simulations of large physical space via coarse-graining. Many QC formulations and methods have been proposed, analyzed, and utilized for the modeling of various physical phenomena. Some formulations differ qualitatively (force-based versus energy-based) and some algorithmically (local/nonlocal treatment). Most simulations presented in the literature are of systems of relatively few degrees of freedom (fewer than one million) and fall short of QC's potential to model large systems of scientific interest. Lastly, many QC formulations suffer from the presence of spurious forces, which are commonly used to cast doubt on the validity of the QC method or the results of particular QC simulations. The goal of this current work was to re-formulate the energy-based QC method in a way that vastly reduces spurious forces and enables QC's full potential for large-scale systems.

5.1 Achievements

The following achievements summarize the unique advancements presented by the current work:

- We have created a novel, fully-nonlocal formulation of the QC method which requires no differentiation between regions (i.e., only one solver is used and no local/nonlocal criterion is necessary). The formulation is energy-based to provide conservative forces and avoid the drawbacks of force-based strategies.
- We separated energy sampling from kinematics by introducing a general summation rule framework through which legacy and novel summation rules are supported. This framework allows us to directly compare the performance of the various sampling methods analytically and numerically.
- Drawing from insights obtained through analysis, we proposed a novel, “optimal” summation rule of first and second order in one, two, and three dimensions. This novel summation accounts for each class of energetically-similar atomic sites produced by the affine interpolation.
- To predict expected accuracy of the summation rules, we analyzed their energy and force errors in one and two dimensions and derived associated error expressions for the large-element limit. The optimal summation rule was predicted to outperform existing summation rules in all metrics.
- After we predicted accuracy, we numerically investigated energy and force errors in two and three dimensions and verified the superior performance of the new summation rules for particular deformation states (without equilibrating or running a solver).
- Integral tests (with equilibration) were then performed on a system with a void and errors with respect to geometrically-equivalent lattice statics simulations were quantified for position and energy errors. The novel summation rules performed phenomenally, with very low errors throughout the entire domain.
- To demonstrate the power of the various summation rules to produce physically correct results, we examined their ability to reproduce physical elastic constants with pathologically-poor meshes.
- The formation and growth of microstructure are of particular interest and we explored dislocation emission and patterns in two- and three-dimensional nanoindentation simulations.
- Due to the characteristic ability of this new QC method to capture atomic effects at surfaces without requiring full atomistic resolution, we verified that the optimal summation rules better capture effects from free surfaces by examining surface energies, the elastic moduli of nanorods, and uniaxial extension of a hole in a nanofilm. None of these coarse-grained examinations are possible without the second-order summation rule introduced in this work.

- We designed and implemented a new, high-quality, modular, and performant realization of the QC method. The implementation uses techniques from high-performance computing to maximize its usage of a single computer, and the algorithm is formulated for distributed-memory parallelism to enable the simulation of arbitrarily-large samples.
- In order to support arbitrarily-high levels of deformation, a novel adaptive neighborhoods algorithm was designed and implemented. This neighborhoods algorithm helps ensure that energetic calculations are correct in the presence of large local rearrangements of atoms and enables simulations with significant plasticity.
- The implementation includes the capability to concurrently simulate multiple grains and allows for the study of large polycrystal domains. This capability will be leveraged in upcoming studies.
- The new QC code was used to simulate nanoscale deformation and failure mechanisms in fcc metals, specifically Cu and Al single- and polycrystals. Results include the microstructural evolution both quasistatically and dynamically.

5.2 Future challenges

The existing formulation and implementation are innovative and, as we have demonstrated, enable accurate analysis of large-scale systems. However, there is room for improvement and the following directions would improve not only the fidelity of possible simulations but also expand the capabilities of this method to simulations of greater scientific interest:

- Adaptive model refinement is a key component of the QC method, and a global form of remeshing (which requires a single computer to store the entire mesh) is currently supported. However, this style of remeshing is slow and wasteful, and a more localized approach is preferred. However, due to this work's unique parallel formulation in which no single computer must store the entire domain, it is difficult to apply most of the remeshing schemes common in finite element literature. A version of local refinement that minimizes communication was implemented, but does not produce meshes of satisfactory quality.

The facts that instances of the program do not store the entire mesh as well as the need to only insert new mesh vertices on valid atomic sites complicate the remeshing process. An elegant solution to the problem may be to disregard the existing connectivity in regions around elements to be refined, insert new vertices, and reconnect the entire subregion. This requires the implementation of constrained Delaunay, advancing-front, or another type of triangulation, customized for the case discrete mesh vertices. This possibility would produce high-quality

meshes and have bounded communication requirements. Another possibility is to adapt known parallel triangulation methods to the case where no single computer has enough information to coordinate and delegate work. Local refinement may be performed by manual bisection to account for the discrete positions of candidate new vertices. Lastly, the existing technique which produces meshes of poor quality may be retained and topological optimization methods can be applied to improve mesh quality without having to perform difficult triangulations.

A performant local remeshing scheme that functions in the case of a fully-distributed mesh is the single most important improvement to make to the current QC implementation. Though the size of the initial mesh would still be limited by the capability of the mesher (to around the low tens of millions of repatoms), simulations of an arbitrary number of repatoms could be performed by adding resolution wherever needed. The added resolution will cause load balancing problems where some instances of the program contain a disproportionate amount of work to perform, but many tools exist to perform parallel load balancing.

- All examples shown here are obtained from equilibration or dynamic evolution at zero temperature. The extension to finite temperature, for example through the maximum-entropy approach of [Venturini et al., 2014], can be added to the fully-nonlocal QC formulation. In fact, the chosen energy-based formulation preserves the Hamiltonian structures and is therefore ideal for this extension. Note that, depending on the specific formulation chosen, the simulation of heat and mass transfer at the nanoscale [Venturini et al., 2014] may also suffer from flux artifacts comparable to force artifacts, which can be analyzed and/or mitigated by the presented QC formulation.
- Much of the difficulty in the implementation of this formulation stems from the fact that sampling atoms, neighborhoods, and remeshing are found or performed in the undeformed configuration. This dependence on the reference configuration forces the creation of the adaptive neighborhoods scheme and imposes constraints on the capabilities and adaptivity of the method. For example, the requirement to have a reference configuration which represents a correct Bravais lattice makes the study of scenarios with pre-existing dislocations difficult. As another example, dislocations require atomistic resolution but once a dislocation has passed through a particular region of the mesh, the resulting repatom positions almost form a Bravais lattice once again. Despite the regular arrangement of repatom positions in the deformed configuration, the deformed mesh is of very poor quality because it uses the connectivity from the undeformed configuration. If the formulation did not require the original reference configuration, the deformed mesh could be locally coarsened when atomistic resolution is no longer needed and the only atomistic resolution necessary to capture the propagation of dislocations would be a pocket around the dislocation.

This formulation, which we term the *Updated Lagrangian Formulation*, would enable unprecedented simulations of extreme efficiency and agility; dislocations could propagate through the entire domain without leaving a wake of expensive and unnecessary atomistic. One of the difficulties in implementing such a formulation is remeshing. If every element in the domain has a distinct set of Bravais lattice vectors, the Bravais lattice vectors of new elements inserted by remeshing must be determined. The set of Bravais lattice vectors assigned to a particular element must be compatible with the vertices of the element, which requires approximation and may result in unphysical meshes in the transition down to atomistic resolution.

Overall, we have contributed a major improvement to the QC community. The two obstacles of spurious forces and free surfaces have been significantly improved within a fully-nonlocal framework that allows for organic incorporation of atomistic resolution into the method without separate regions or differing treatments. We have demonstrated the superior performance of this new framework in analysis as well as numerical experiment, in particular configurations as well as integral tests, and in bulk behavior as well as free surfaces. We have presented a performant, high-quality implementation of the formulation and enumerated the key components required for the community to incorporate our work into their own. There remain improvements to be made, but the present work not only demonstrates capabilities and capacities greater than existing methods but also lays the foundation for multiple directions of unprecedented and unparalleled future work within the Kochmann research group on the forefront of scientific research.

Appendix A

High-deformation adaptive neighborhoods

A.1 Overview

The degrees of freedom in the QC method are the repatom positions. In the specific fully-nonlocal formulation of this work, energetic quantities are sampled at sampling atom locations, which are obtained via interpolation of the repatom positions. However, in order to calculate sampling atom energies, repatom forces, sampling atom centrosymmetries, or any other property related to sampling atoms, a representation of the local atomic neighborhood of every sampling atom is needed. As with the sampling atoms themselves, these neighbor positions are also dependent on the locations of repatoms. At a high level, the purpose of the initialization phase of the simulation is to populate sampling atoms and set up their neighborhoods, and the solution phase of the simulation simply iterates through the neighborhoods many times to update neighbor positions, calculate repatom forces, and advance repatom positions.

It is crucial that, whenever energetic quantities are calculated at sampling atoms, those sampling atoms consider all of the lattice sites which would be within the potential's cutoff radius if the simulation were performed with MD. In simulations that experience purely elastic deformation, the neighbors of a given sampling atom in the beginning of the simulation will most likely be sufficient and no further neighbors need to be found. However, in simulations with plastic deformation where there exists large relative atomic motion, sampling atoms will require updates of their local atomic neighborhoods to account for initially-distant neighbors that gradually move into their immediate neighborhood.

For an example, see Figure [A.1](#), which is a still from a movie available [here](#). The scenario is 2D spherical nanoindentation using the Extended Finnis Sinclair potential for copper; however, the simulation is run for illustration of the neighborhoods algorithm and not for its physical relevance. In particular, dislocations form and propagate long before the snapshot shown in Figure [A.1](#). In

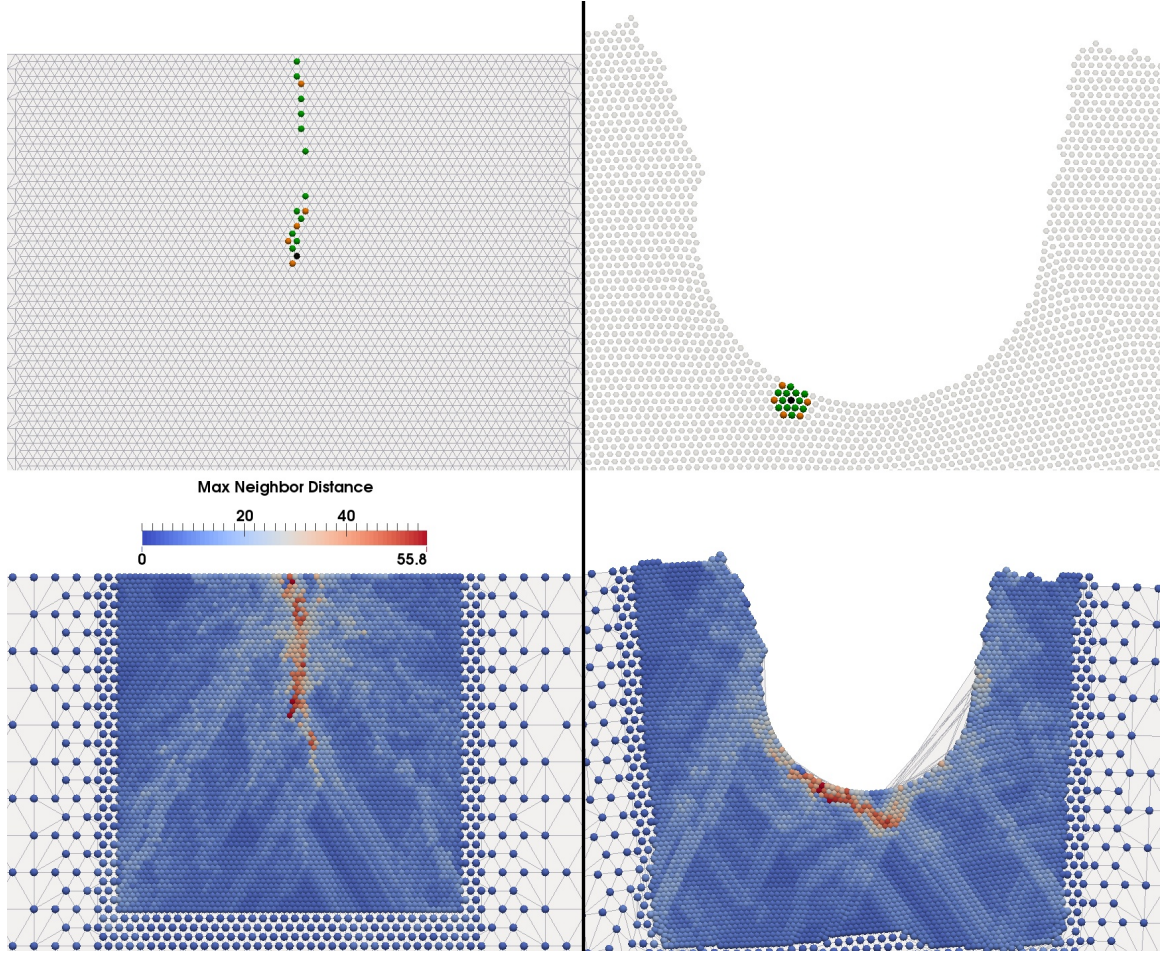


Figure A.1: A still from a severely-deforming 2D nanoindentation example made to illustrate the adaptive neighborhoods algorithm. The left two panels are drawn in the undeformed configuration, the right two panels are in the deformed configuration. The upper two panels show the neighborhood of a given sampling atom in the undeformed and deformed configurations; green spheres are neighbors within the potential’s interaction distance and orange spheres are not within the interaction distance. Over the course of the simulation, some sampling atoms need neighbors in the deformed configuration were originally far away in the undeformed configuration, and the lower two panels color all sampling atoms by the distance to the farthest neighbor in the undeformed configuration. In particular, all of the sampling atoms colored red in the lower panels indicate that one of their current neighbors in the deformed configuration is found more than 50 Å away in the undeformed configuration. The deformation is sufficiently high that, for many sampling atoms, none of their starting neighbors remain neighbors by the end. The full movie is available [here](#).

order to measure meaningful physical quantities from the simulation, remeshing is needed to allow the dislocations to propagate through the domain or the results will be corrupted by the entrapment of the dislocations. However, in the shown example we have disabled remeshing and applied severe deformation in order to test the neighborhood algorithm.

A neighborhoods algorithm capable of supporting such high deformation must ensure that each sampling atom has all of the neighbors it needs in the deformed configuration at any given time.

However, it must also not include too many neighbors, or the simulation will develop a prohibitively large memory footprint and neighbor calculations will become far more expensive than what is strictly required for correctness. In the following sections we describe the implemented neighborhood algorithm, which attempts to achieve high performance while supporting arbitrarily large deformations.

Note that details regarding the non-trivial interaction between the neighborhoods algorithm and parallelization are addressed in Section C.4

A.2 What information is stored

In molecular dynamics, atomic neighborhoods are obtained for a given timestep by performing neighbor searches among all of the atoms in the simulation. In the QC formulation of this work, no such neighbor searches can be performed because there is no pool of “available neighbors” in which to search. While it is true that in regions of atomistic refinement, the neighbors of a sampling atom will be repatoms and one couple perform neighbor searches on the repatoms. However, in regions of any coarsening, the neighbors of a sampling atom will be neither repatoms nor sampling atoms - they are simply points which are linearly interpolated from repatom positions. A sampling atom’s “neighbor”, therefore, is a tuple of “repatom dependencies”, where a repatom dependence is a pair of a repatom’s index and a corresponding shape function value.

To calculate the position of a neighbor, one must only iterate through the neighbor’s repatom dependencies and multiply the position of each referenced repatom by the corresponding shape function value. Changing between the *undeformed* and *deformed* configurations is accomplished by using either the mesh vertices (undeformed repatom positions) or the deformed repatom locations as the positions used in interpolation.

If a neighbor lies on a repatom (on a vertex of the mesh), then only one repatom dependence is needed to represent it, with a shape function value of 1. If a neighbor lies on or within an element, then up to `SpatialDimension+1` repatom dependencies are needed to represent it (i.e., up to 3 dependencies in 2D and 4 dependencies in 3D). The sum of the shape function values found in all of the repatom dependencies for a given neighbor must be 1; however, individual components do not have to be strictly positive (this is discussed in Section 2.3.2.5 to allow for Cauchy-Born-type sampling atoms).

The absolute minimum that the container of sampling atoms neighbors needs to store is all of the repatom dependencies (between 1 and 4) for all of the neighbors (typically between 1 and 100) of all of the sampling atoms (typically between 10^3 and 10^7). However, the positions of, unit vectors to, and distances to neighbors are used by several sections of the simulation on each iteration (such as computing centrosymmetry and determining if any neighbors pass the relative movement trigger

described in Section [A.3.3](#)), so neighbor positions, distances, and unit vectors are stored as well. While it would be possible to not store these quantities and recompute them each time they are needed, doing so is detrimental to performance because it re-incurs all of the cache misses required to compute the position of the neighbor by reading from potentially-uncached repatom positions.

A.3 Ensuring necessary neighborhoods under large deformation

If, during the course of a loadstep, a sampling atom does not have in its neighborhood a lattice site that has entered the potential’s interaction distance, the equilibrated solution is incorrect. When sampling atoms do not have all the required neighbors, it is common for them (or the repatoms they depend on) to collapse onto the same location, move too close to each other, or produce other unphysical results.

A.3.1 Goal and purpose

For a neighborhood scheme to be correct, it must guarantee that each sampling atom always has all of the neighbor atoms that are within the potential’s interaction distance. The most correct, but least efficient, method of guaranteeing that each sampling atom has all of its required neighbor atoms is for each sampling atom to have in its neighborhood every single lattice site in the entire domain. The goal of the adaptive neighborhoods is to obtain exactly the same answer as one would obtain with this expensive version while using the smallest neighborhoods possible.

A.3.2 High-level view of neighborhood generation

Instead of storing each and every lattice site in the entire domain, each sampling atom stores only the lattice sites that are within a certain `deformedCutoffRadius` *in the deformed configuration*. However, in the QC formulation of this work neighbors cannot be generated in the deformed configuration but must instead be generated in the undeformed configuration, mapped to the deformed configuration using the current repatom positions, and then checked for proximity to the sampling atom. The conceptual algorithm for generating sampling atom neighborhoods is shown in Listing [A.1](#).

Listing A.1: Conceptual algorithm for generating sampling atom neighborhoods

```

for each sampling atom
  candidateNeighbors = generate all lattice sites within some
    undeformedCutoffRadius in the undeformed configuration
  map candidateNeighbors to the deformed configuration
  neighbors = []
  for each candidateNeighbor in candidateNeighbors
    map candidateNeighbor to deformed configuration
    if deformed distance to candidateNeighbor is within the deformedCutoffRadius
      add candidateNeighbor to neighbors

```

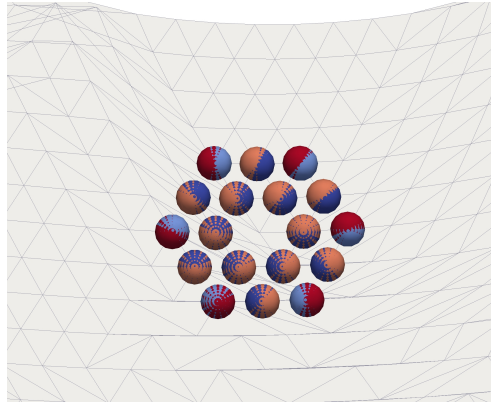
A.3.3 When to rebuild

Once a neighborhood has been built and then reatom positions are subsequently moved (through a loadstep iteration’s update or by any other means), it is difficult to know whether a given sampling atom still has a complete neighborhood. If the new reatom positions do not bring any new lattice sites within the potential’s interaction distance of the sampling atom, then the existing neighborhood is sufficient. However, if the new reatom positions do bring a new lattice site within the cutoff radius, the sampling atom’s neighborhood must be rebuilt.

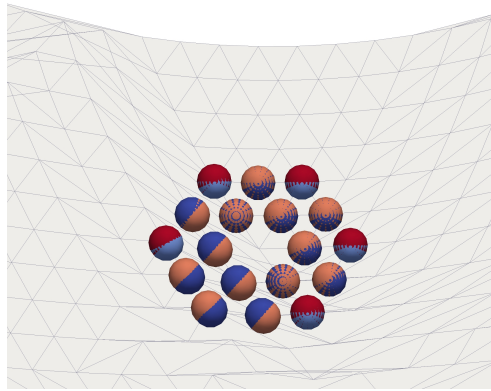
The metric chosen for determining whether a sampling atom’s neighborhood needs to be rebuilt is designed in the spirit that, if a new lattice site is entering the neighborhood, the existing neighbors must be “moved out of the way.” Every time a sampling atom’s neighborhood is built, the relative position of each neighbor of the sampling atom is stored. To determine whether a given sampling atom’s neighborhood is still sufficient for a new deformed configuration, the updated relative position of each neighbor is calculated. If the relative movement of a neighbor of a sampling atom is sufficiently large, the sampling atom’s neighborhood needs to be rebuilt. The relative movement tolerance is arbitrary, but values of around 0.3 Å have performed well in numerical experiments.

This algorithm is illustrated in Figure A.2 and can be watched [here](#). In order to reduce the cost of neighborhood updates, neighbors which are found to be sufficiently far away from the sampling atom but within the deformed cutoff radius are marked as “backup” neighbors. These backup neighbors are in the neighborhood but are far enough away that they are updated and checked less frequently than the others until they become sufficiently close. In Figure A.2, the orange/dark blue neighbors are neighbors that are always updated and used in calculations, while the light blue/red neighbors are backup neighbors.

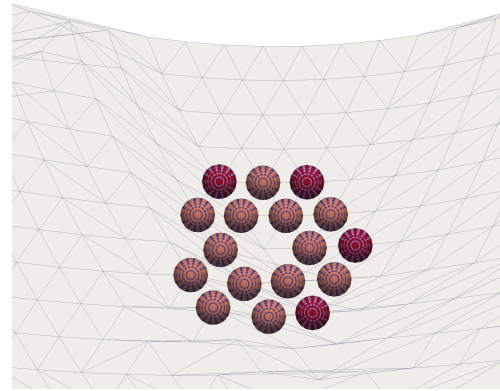
Because rebuilding even one neighborhood out of the entire set of neighborhoods has some



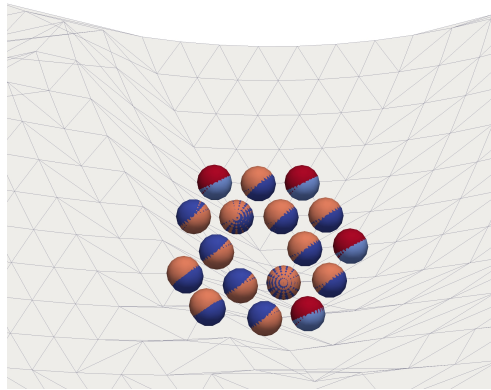
(a) Beginning of the loadstep



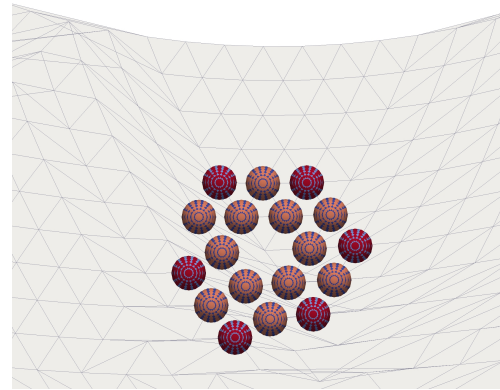
(b) On some iteration, a neighbor passes the relative movement trigger and flags the neighborhood to be rebuilt



(c) The sampling atom's neighborhood is rebuilt, dropping one of the neighbors.



(d) Some iterations later, the neighborhoods have moved enough to trigger a rebuild.



(e) The sampling atom's neighborhood is rebuilt again, taking into account a new neighbor.

Figure A.2: An example of the high deformation neighborhood algorithm rebuilding a neighborhood. Red and orange spheres are the current positions of neighbors. Light and dark blue spheres are the relative positions of those neighbors at the last build. When a red or orange sphere overlaps significantly with a blue sphere (such as in the third panel), the neighbor has not moved much since the last neighborhood build. A movie is available [here](#).

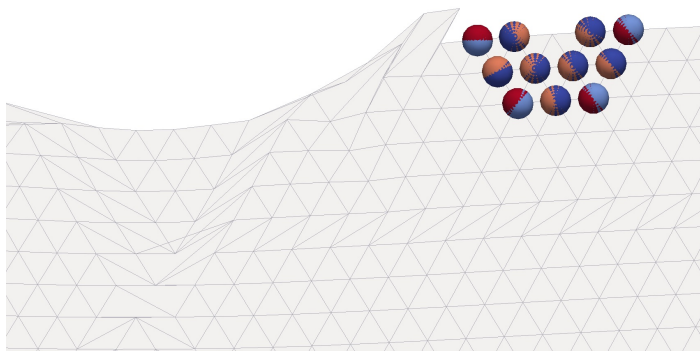
fixed cost related to global data structures (element incidences, setting up data structures to find closest simplices, etc.), it is preferable to build neighborhoods in larger batches rather than, for example, once per solver iteration. Therefore, the algorithm is currently configured such that, if on a given iteration any sampling atoms are flagged using the relative movement tolerance, all sampling atoms that are close to the tolerance are rebuilt as well. “Close to” means that their relative neighbor movement is at least some threshold times the relative movement tolerance. Initial numerical experimentation indicates that a threshold of around 0.5 gives a good trade-off between processing sampling atoms in batches that are reasonably sized and rebuilding more sampling atoms than are perhaps strictly necessary.

It is important to note that we are using the relative movement *of the neighbors we know about and are tracking* as a proxy for whether other lattice sites *that we do not know about, and are not tracking* might have entered the neighborhood. There is no guarantee that this proxy will ensure that the neighborhood is actually correct. If there is a situation in which an untracked lattice site (which is not a neighbor) enters the neighborhood and yet none of the tracked neighbors are moving sufficiently because of it, this algorithm will fail to rebuild the neighborhood and perhaps allow for an incorrect neighborhood. This is a difficult case to capture: if none of the neighbors of a sampling atom are moving very much, how can the sampling atom know that it needs to start tracking a new neighbor?

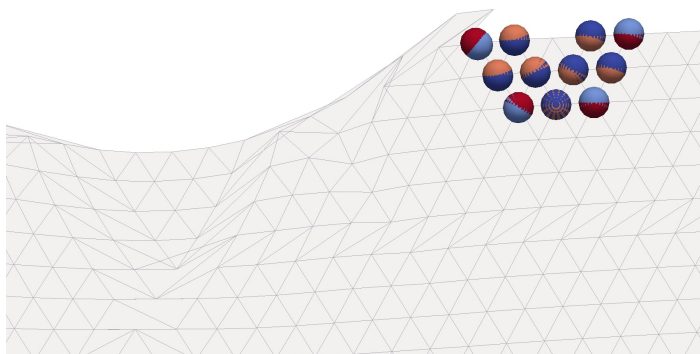
One example of this type of pathological behavior is on the surface of nanoindentation and is shown in Figure A.3 (the coloring follows the same coloring scheme described above for Figure A.2) and can be watched [here](#). A sampling atom on the surface sees a relatively calm neighborhood, and yet atoms that curl away from the indenter can approach surface atoms without disturbing the neighborhood of those surface atoms. The figure shows a particular case of a general situation in which a neighborhood starts correct at the beginning of the loadstep (the first panel), a lattice site moves within the neighborhood undetected because the algorithm does not detect any reason to rebuild the neighborhood (the second panel), and finally the new lattice site is detected and included in the neighborhood after some other neighbor moves enough to cause a neighborhood rebuild (the third panel).

A.3.3.1 Example failure rate

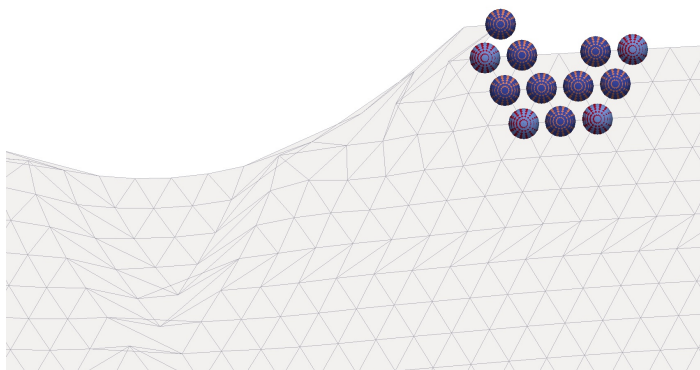
The performance and correctness of the adaptive neighborhoods algorithm can be measured and tracked. Whenever a sampling atom’s neighborhood is rebuilt, the number of neighbors within the potential’s interaction distance is measured before and after the rebuild. If the number of neighbors within the potential’s interaction distance increases during the rebuild, then the previous neighborhood was incorrect. The code keeps track of and reports the frequency of this type of error over the course of a loadstep, which is not considered fatal unless the incorrect neighborhood is



(a) Neighborhood at the beginning of the loadstep



(b) A lattice site shifts into the radius, but the neighbors haven't moved enough to trigger a rebuild



(c) Eventually, the neighbors move enough to trigger a rebuild, when it is discovered that the neighborhood was incorrect

Figure A.3: An example in which the neighborhood algorithm fails. Due to the free surface, atoms may approach a surface sampling atom without disturbing the sampling atom's local neighborhood. The color scheme is explained in Figure A.2 and a movie of this process can be seen [here](#).

caught after a loadstep has equilibrated.

For example, over the course of the nanoindentation simulation shown in Figure A.1, the 50-loadstep simulation took 355,000 iterations, for a average of 14,000 iterations per loadstep. 2,800, or 0.79% of the iterations required at least one neighborhood rebuild and a total of 194,000 neighborhoods were rebuilt. Of those, 6 (0.003%) of the rebuilt neighborhoods were erroneous.

A.3.4 Undeformed and deformed cutoff radii

For a given sampling atom, the neighborhood container keeps track of more neighbors than it strictly needs to for correctness, monitors their relative movements to estimate when it needs to rebuild the neighborhood, and rebuilds the neighborhood when a neighbor has moved too much. When the neighborhood container builds a set of candidate neighbors, it keeps those that lie within a certain radius *in the deformed configuration*, the `deformedCutoffRadius` (see Figure A.4). This radius has been chosen as a factor times the relative distance a neighbor moves before the neighborhood is rebuilt. The value chosen for the factor is a function of how drastically the neighborhoods are expected to change for a given scenario, but using a factor of 1 is risky and factors of 2 or 3 are conservative.

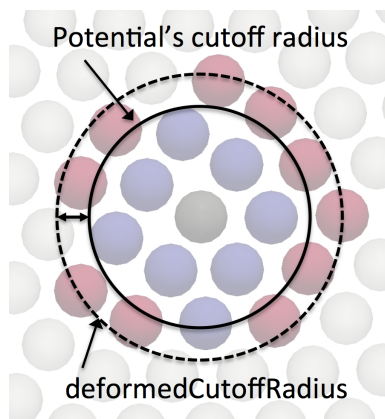


Figure A.4: Neighborhood buffer distance illustration.

While the `deformedCutoffRadius` controls which lattice sites in the deformed configuration are kept in a sampling atom's neighborhood, the candidate pool must first be generated *in the undeformed configuration* and mapped into the deformed configuration. Unfortunately, there is no easy closed form expression for how far away one must generate lattice sites in the undeformed configuration in order to cover all necessary neighbors in the deformed configuration. The general strategy used in the current algorithm is to generate neighbors within a fixed distance beyond the largest distance (in the undeformed configuration) to a current neighbor, assuming that new necessary neighbors will always be within some bounded distance away from existing neighbors.

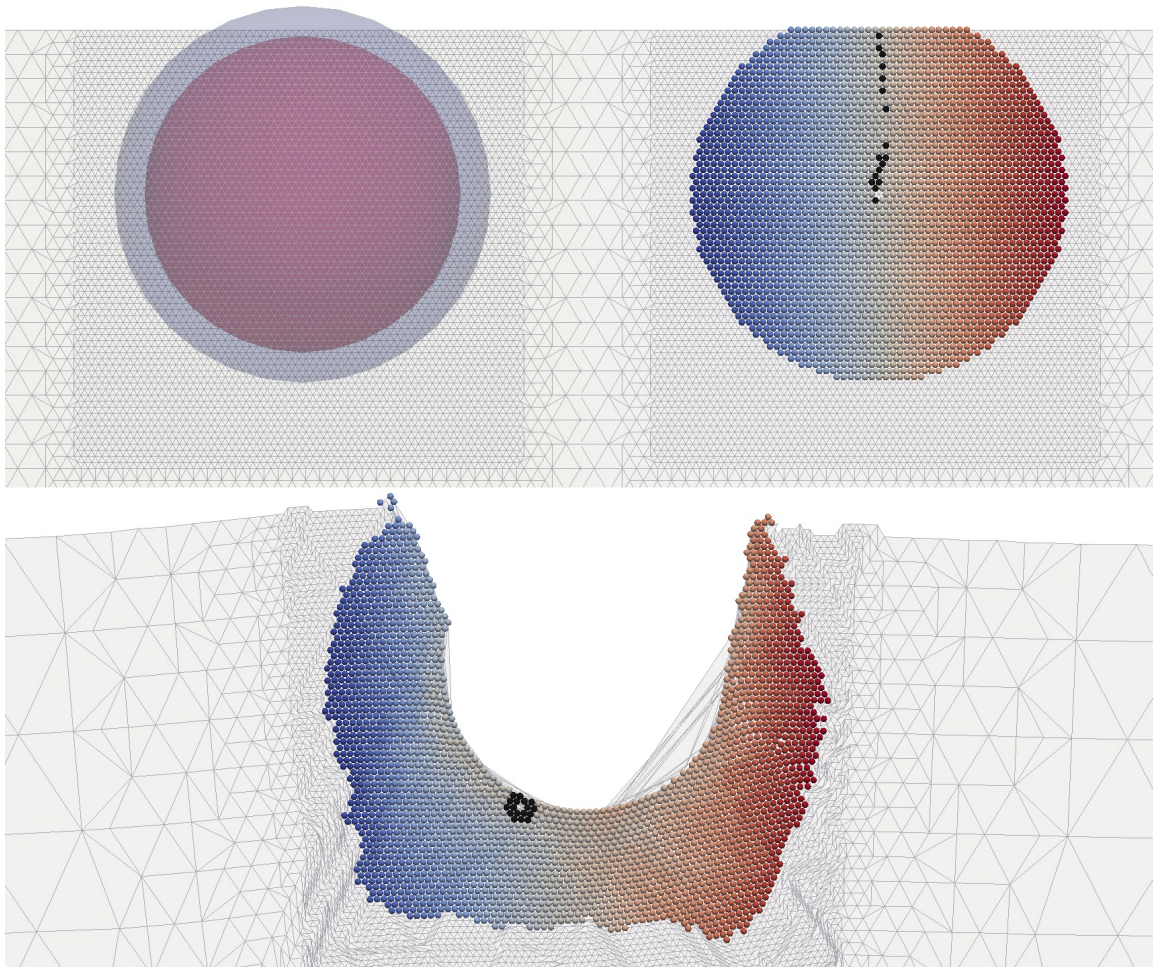


Figure A.5: The neighborhood of a sampling atom experiencing severe deformation. The large red circle in the upper left panel represents the `maxNeighborDistance`, the larger blue circle represents the `undeformedCutoffRadius`. The candidate neighbors are colored arbitrarily from blue to red in the undeformed and deformed configurations, and the retained neighborhood (which is stored after the rebuild) is colored black. The movie of the evolution of this neighborhood can be seen [here](#).

Lattice sites are generated in the undeformed configuration with an `undeformedCutoffRadius`, which is specific to each sampling atom and is a function of how highly-deformed its neighborhood is. Sampling atoms keep track of the largest distance to one of their neighbors in the undeformed configuration, the `maxNeighborDistance`. The `undeformedCutoffRadius` must be larger than the `maxNeighborDistance` or the sampling atom would not be generating any more possible neighbors than it was before, so the `undeformedCutoffRadius` is chosen to be the sampling atom's `maxNeighborDistance` plus some neighborhood expansion distance, which depends on how highly the neighborhoods of a given scenario are expected to deform. In the severe-deformation nanoindentation scenario of Figure A.1, a neighborhood expansion distance of 5 was too small, but 10 was sufficient.

Figures A.5 and A.6 (movie versions [here](#) and [here](#)) illustrate the `maxNeighborDistance`,

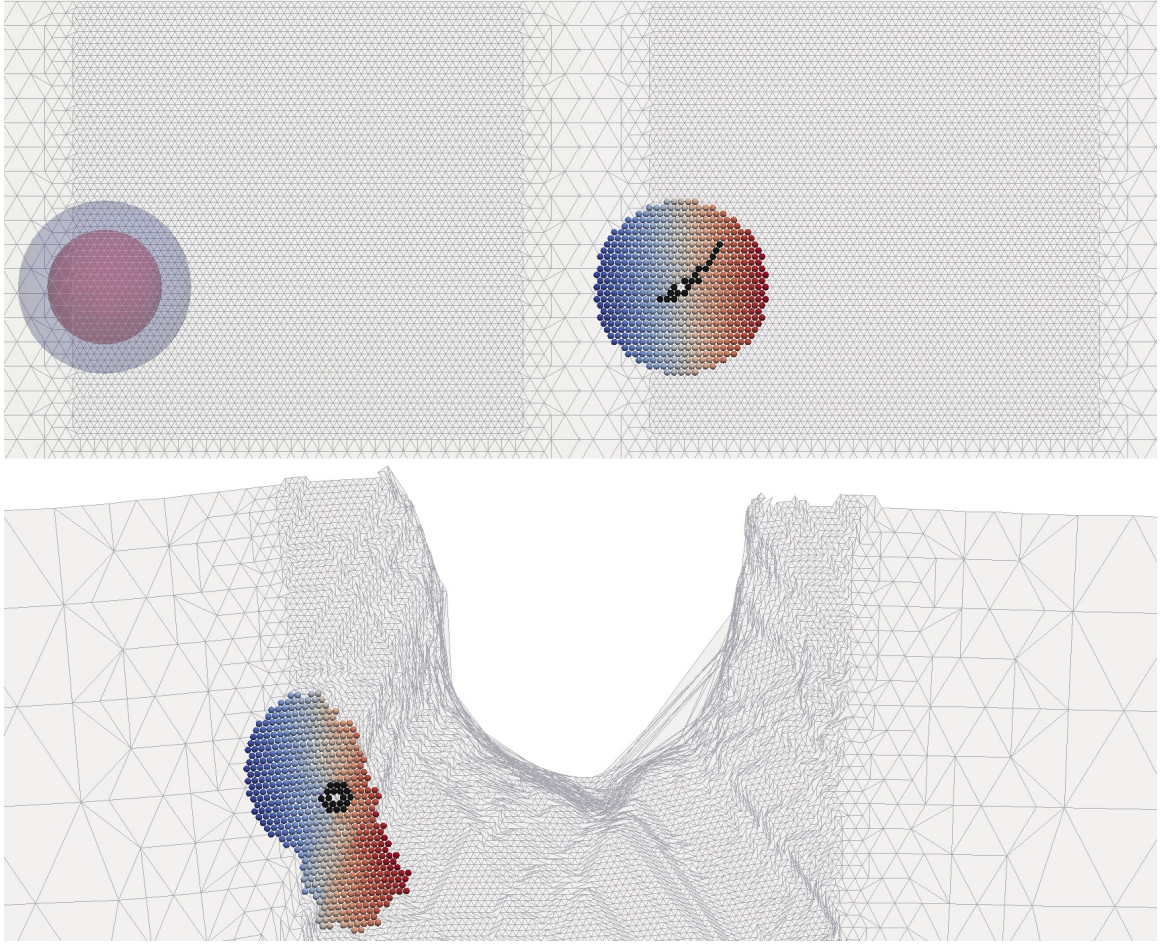


Figure A.6: The neighborhood of a sampling atom experiencing less deformation than that of Figure A.5 and this figure shares the same coloring scheme. The movie of the evolution of this neighborhood can be seen [here](#).

`undeformedCutoffRadius`, the candidate neighbors, the candidates mapped to the deformed configuration, and the actual retained neighborhood. Both figures are from the end of the simulation, but the highly-deforming neighborhood had 393 rebuilds and the less highly-deforming neighborhood had 129 rebuilds. The linked movies show the evolution of the `maxNeighborDistance` over the course of the simulation, which starts low (4.95) and grows to very large (55.8, for Figure A.5). Because of the very large `maxNeighborDistance` achieved of 55.8, many candidate neighbors are generated in the undeformed configuration and mapped to the deformed configuration, and in 3D the number would be much higher. However, it is critical to point out that only a small fraction of those candidates are kept in the neighborhood for use during iterations.

A.4 Neighborhood construction for different sampling rules

In order to support the various legacy and novel summation rules supported in this QC realization, the neighborhood container must be able to build and support neighborhoods for four qualitatively different types of sampling atoms: non-Cauchy-Born-type sampling atoms with neighboring elements with the same orientation, non-Cauchy-Born-type sampling atoms with neighboring elements of different orientations, Cauchy-Born-type sampling atoms inside of an element, and Cauchy-Born-type sampling atoms on the face of an element. The general procedure used to generate the sampling atom neighborhoods is shown in Listing A.2. The sampling atom type affects how the pool of candidate neighbors is generated, but the remainder of the algorithm is the same.

Listing A.2: Algorithm for generating sampling atom neighborhoods

```

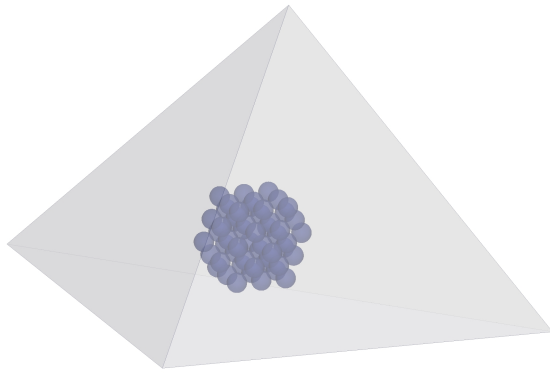
for each sampling atom
  determine this sampling atom's undeformedCutoffRadius
  generate the candidate neighbors according to this sampling atom's type
  for each candidate neighbor
    map candidate to the deformed configuration
    if the candidate's deformed positions is within the deformedCutoffRadius
      add the candidate to the neighborhood

```

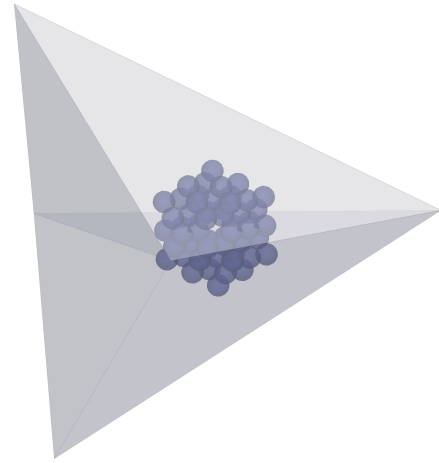
A.4.1 Cauchy-Born-type sampling atoms

The neighbors of Cauchy-Born-type sampling atoms have prescribed deformations applied to them, regardless of in which elements they actually lie in physical space. As discussed in Section 2.3.2.5, this is achieved by allowing the shape function values of neighbor positions to have negative components. As with all sampling atom types, once the candidate neighbors are generated in the undeformed configuration, they are mapped to the deformed configuration and those that lie within the `deformedCutoffRadius` are kept as the sampling atom's neighborhood. Figure A.7 shows examples of neighborhoods for different element sizes for Cauchy-Born-type sampling atoms within elements (first column) and on the faces of elements (second column).

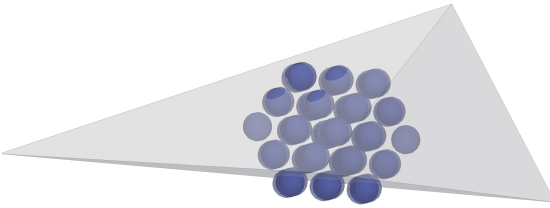
Inside of elements In both first- and second-order optimal summation rules, there are sampling atoms at the barycenter of each (sufficiently-large) element. These sampling atoms “see” only the deformation of their containing element, irrespective of the element size. The approximation is excellent for elements whose ball of interaction around the sampling atom is fully-contained within the element, it is still good for elements where the ball extends beyond the element boundaries, but it is poor when the ball completely contains the element.



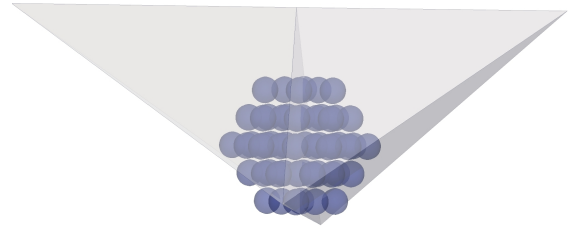
(a) Barycenter Cauchy Born Sampling Atom in a Large Element



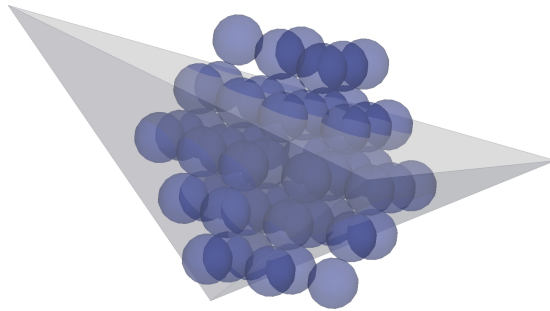
(b) Face Cauchy Born Sampling Atom in Large Elements



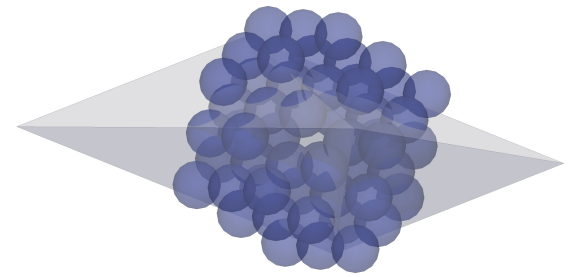
(c) Barycenter Cauchy Born Sampling Atom in a Medium Element



(d) Face Cauchy Born Sampling Atom in Medium Elements



(e) Barycenter Cauchy Born Sampling Atom in a Small Element



(f) Face Cauchy Born Sampling Atom in Small Elements

Figure A.7: Illustrations of Cauchy-Born-type sampling atoms in elements of different sizes.

To generate the neighborhood for a Cauchy-Born-type sampling atom inside an element, one must first identify the containing element and generate candidate neighbors in the undeformed configuration. Whether or not the candidate neighbor lies inside of the element that contains the sampling atom, each candidate neighbor's shape functions values are calculated with respect to the containing element.

On element faces In the second-order optimal summation rule, there are sampling atoms on the (sufficiently-large) faces of each element. Regardless of the sizes of the elements, these sampling atoms “see” the deformation of two elements: half the neighborhood will be deformed according to the gradient of one and half according to the other.

To generate the neighborhood for a Cauchy-Born-type sampling atom on the face of an element, one must first identify its two incident elements and generate candidate neighbors in the undeformed configuration. The two elements are separated by the plane of the face joining them. Regardless of whether the candidate neighbors lie inside either of the two elements, the shape function values for the candidate neighbors are calculated with respect to the element on whichever side of the dividing plane the neighbor lies.

A.4.2 Non-Cauchy-Born-type sampling atoms in regions of uniform crystal orientation

None of the sampling atoms in quadrature and cluster sampling have Cauchy-Born deformation, and neither do the sampling atoms on repatoms in the optimal rules. These sampling atoms “see” the deformation of all incident elements.

To generate the neighborhood for a non-Cauchy-Born-type sampling atom in a region of uniform orientation, one first generates lattice sites up to the `undeformedCutoffRadius` in the undeformed configuration using the orientation of the region and the sampling atom’s undeformed position as an existing lattice site. In order to calculate the shape function values of these candidate neighbors, the containing simplex must be found for each candidate, which is an expensive process. If a candidate neighbor has no containing simplex, it is outside of the mesh and is discarded. The shape function values of the remaining candidates are calculated with respect to their containing simplices, and no negative shape function values are permitted. Candidate neighbors are mapped to the deformed configuration, and those that lie within the `deformedCutoffRadius` are kept as the sampling atom’s neighborhood. Figure A.5 shows an example of how this type of neighborhood is constructed.

Listing A.3: Neighborhoods for polycrystals

```

for each element incident on the ball of radius undeformedCutoffRadius
  if the element has no orientation
    add the vertices as candidate neighbors
  else
    generate possible sites with this element’s orientation
    for each possible site
      if the possible site is within this element
        add possible site to candidate neighbors

```

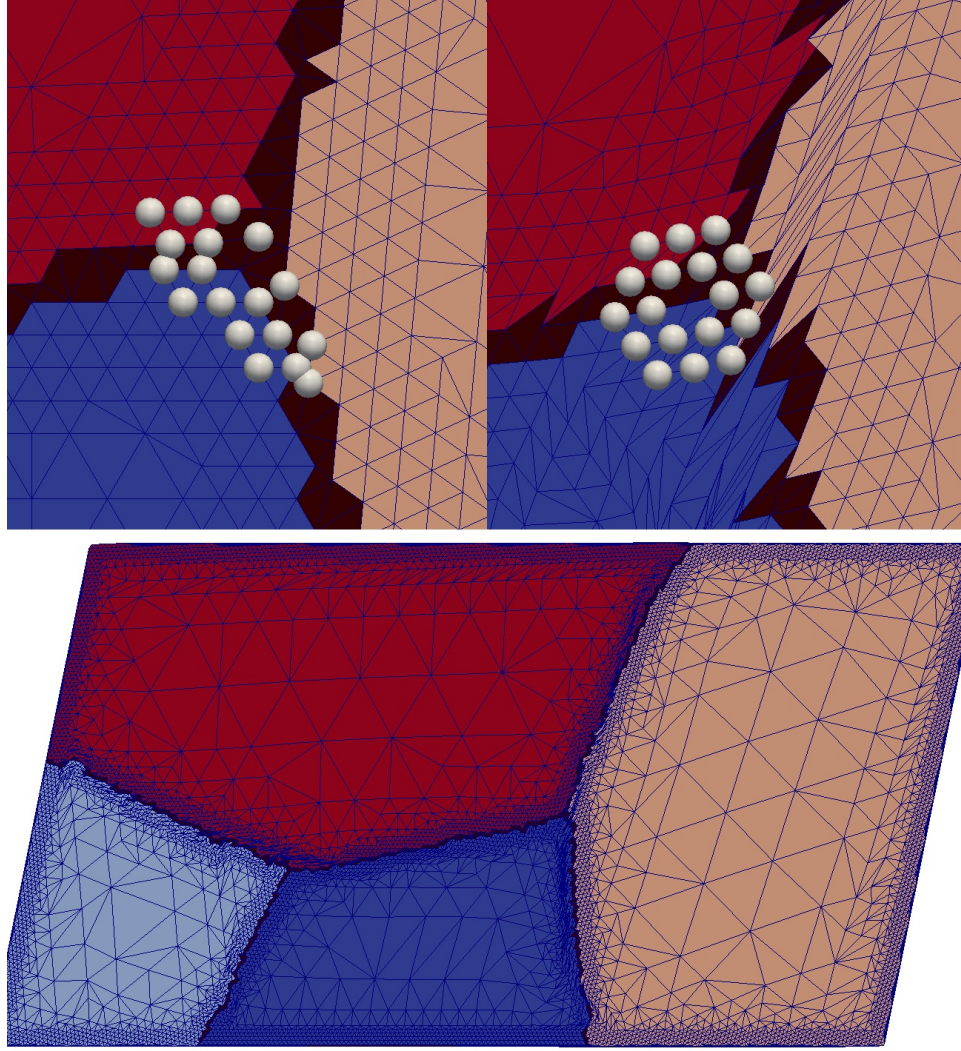



Figure A.8: An example of neighborhoods for polycrystal simulations. The upper left panel shows the neighborhood in the undeformed configuration, the upper right panel shows the deformed configuration. The bottom panel shows the deformation of the entire domain. Elements are colored by their orientation. A movie is available [here](#).

A.4.3 Non-Cauchy-Born-type sampling atoms in polycrystal simulations

In polycrystal simulations, each element in the simulation has a specified orientation, and elements on grain boundaries that have no orientation receive an indicative marker. When the elements incident on a sampling atom are all atomistic, only their nodes are added as neighbors. However, when incident elements are not atomistic and have different orientations (i.e., near a grain boundary), care must be taken to generate correct lattice sites.

That is, any sampling atom whose ball of radius `undeformedCutoffRadius` is incident on elements of different orientations must be sure to generate only valid lattice sites with respect to the orientation of each element. Because each element can (in principle) have a different orientation,

the generation of neighborhood lattice sites can be accomplished as an operation over all incident elements (see Listing [A.3](#)).

In 3D simulations experiencing significant deformation, sampling atoms can be incident on thousands of elements and the algorithm of Listing [A.3](#) is too costly. Most sampling atoms are incident on elements of at most a few orientations, and in practice this fact is leveraged to greatly reduce the cost of generating neighborhoods by performing a few distinct generations of uniform orientation (Section [A.4.2](#)) and not performing the full general algorithm of Listing [A.3](#).

An example of a 3D polycrystal simulation can be seen in Figure [4.16](#), though it is too difficult to illustrate neighborhoods with different orientations in 3D. Figure [A.8](#) shows neighborhoods for a 2D polycrystal simulation undergoing simple shear, and a movie of the simulation is available [here](#).

Appendix B

Repairing meshes

B.1 Overview

The mesh connecting the undeformed repatom positions is a key component of the simulation, affecting sampling atom positions, neighbor positions, adaptive model refinement, and many other aspects. Except for the exceptional case when a mesh is a uniformly coarsened box and meshing can be done manually (see, for example, free surface simulations in Section 3.5 where uniform meshing is necessary to avoid nonuniform surface forces), external software packages are used to mesh the convex hull of the undeformed repatom positions as well as any provided vacant locations (see section 4.3.1). In two dimensions, both `triangle` [Shewchuk, 1996] and `CGAL` [Hert and Seel, 2015] were used and exhibited no difficulties. Meshing in three dimensions has proven to be a significantly more challenging task.

B.2 Three dimensional meshing and degenerate elements

In general, three dimensional Delaunay triangulation is made difficult and the solution becomes non-unique when input points are co-spherical because it is challenging to perform the incircle or insphere tests used in intermediate triangulation steps. Unfortunately, the regular arrangement of atomic positions in a QC mesh has many instances of co-spherical points and poses great difficulty for meshers. For example, Vtk’s mesher simply does not mesh almost any QC mesh because it cannot solve required linear systems.

In three dimensions, we have tested many different meshers (Vtk [Shroeder et al., 2006], hull [Clarkson et al., 1993], Qhull [Barber et al., 1996], Tetgen [Si., 2015], CGAL [Hert and Seel, 2015], Matlab, and Mathematica), but all tested meshers produce meshes with problematic or invalid elements, primarily zero-volume tetrahedra. Returning to the Vtk mesher’s inability to complete meshes, it is possible to coax it to mesh by perturbing the positions of the input points (what qhull calls “Joggling the input”). Though this causes Vtk to complete the meshing process, perturbing the input

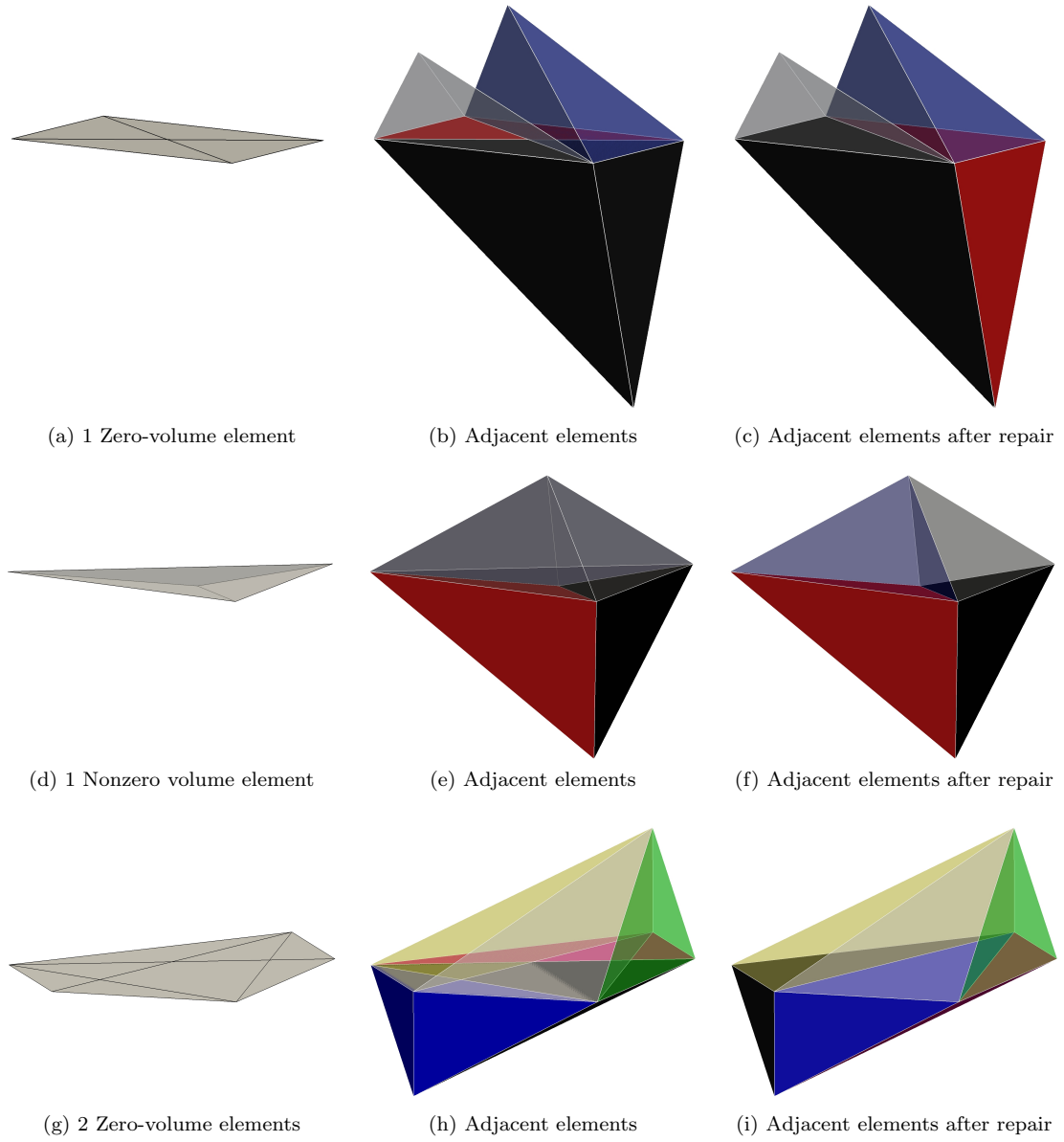


Figure B.1: Zero-volume element repair of two adjacent degenerate tetrahedra.

points magnifies the problem of zero-volume tetrahedra. If input points are perturbed, the meshers mesh the perturbed points and create many elements of very small volume and high condition number. When the connectivity from the mesh of the perturbed points is applied to the original and unperturbed points, the elements of very small volume become zero-volume elements.

Vtk is the only mesher for which we perturb input points for the purpose of coaxing it to solve, and unsurprisingly this produces many zero-volume (less than $1 \times 10^{-6} \text{\AA}^3$) elements. If these same perturbed input points are given to the other meshers, many zero-volume elements are produced. However, in practice we do not perturb the input points to the other meshers because it is not

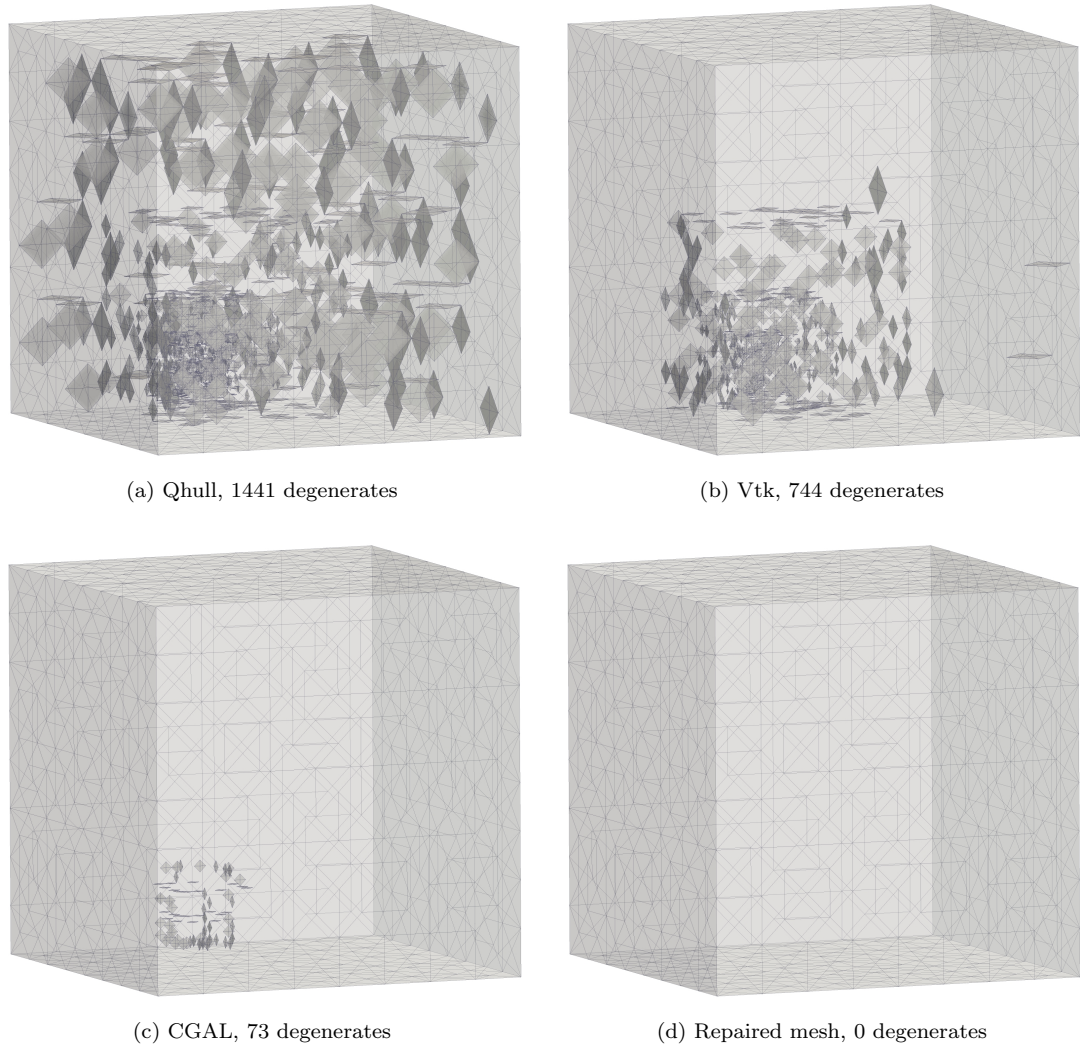


Figure B.2: Repairing the sharp mesh from the Interface scenario 3.2.3.

necessary in order for them to perform triangulation. Nevertheless, despite the fact that the input points are unperturbed, the other meshers themselves produce zero-volume degenerate tetrahedron, examples of which can be seen in Figure B.1 for an isolated degenerate tetrahedra and a pair of adjacent degenerate tetrahedra. These degenerate tetrahedra disrupt many processes in a QC simulation, such as generation of sampling atoms and adaptive model refinement.

In particular, consider the case of second-order optimal sampling atom creation, where sampling atoms are created on the faces of elements. These side sampling atoms have neighbors that experience either the deformation gradient of one side or the other side of the face. When there are degenerate tetrahedra, the deformation gradient to use for one half of the neighbors of a sampling atom on an element side is not well defined: the deformation gradient for the degenerate tetrahedron is

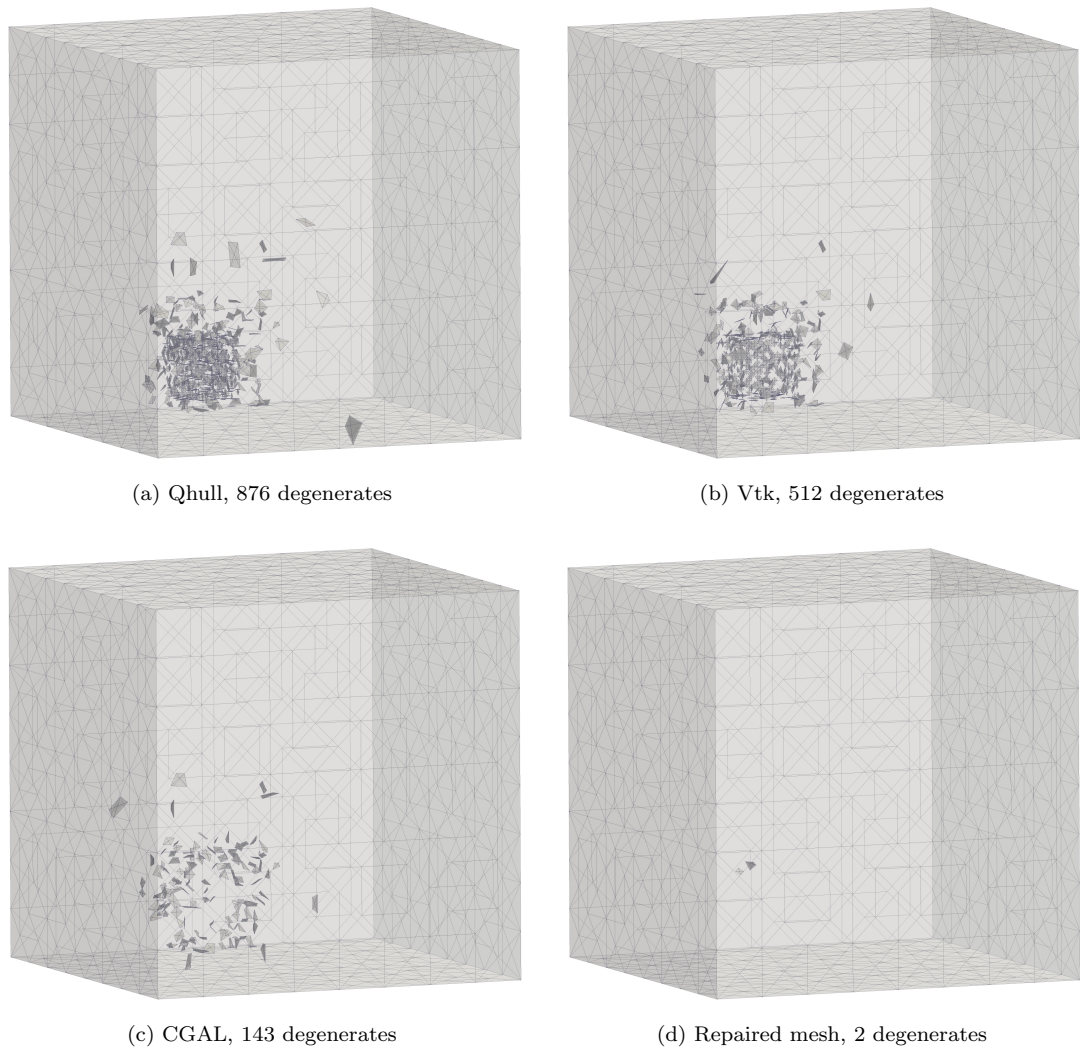


Figure B.3: Repairing the diffuse mesh from the Interface scenario [3.2.3](#).

pathological and there are two elements of non-zero volume that are physically proximal to the side. In the current implementation, no side sampling atoms can be made for faces of elements that are adjacent to degenerate tetrahedra.

As a second example of the difficulties that zero-volume tetrahedra present to the QC simulation, consider adaptive model refinement, or “remeshing”. Roughly speaking, it is impossible for refinement to “pass through” the barrier of a degenerate element. Because the deformation gradient of the degenerate element is pathological, it is ignored by the refinement criterion and the element is therefore never chosen to be refined. If the degenerate elements are removed from the mesh, then the mesh has a topological, though not volumetric, hole through which mesh refinement cannot propagate. A typical deformation mechanism in QC simulations is slip, where an entire sheet of

	Sharp 3d Interfaces B.2	Diffuse 3d Interfaces B.3	Elastic Constants B.6	Two Voids B.5
Qhull	1441	876	902	15547
Vtk	744	512	958	4393
Tetgen	72	141	559	3476
CGAL	73	143	569	3486
Matlab	0	116	724	3575
Mathematica	0	117	711	3586
Repaired qhull	0 (100%)	2 (99.77%)	9 (99.00%)	248 (98.40%)
Repaired vtk	0 (100%)	3 (99.41%)	15 (98.43%)	199 (95.47%)
Repaired tetgen	0 (100%)	2 (98.58%)	10 (98.21%)	208 (94.01%)
Repaired CGAL	0 (100%)	2 (98.60%)	8 (98.59%)	190 (94.54%)

Table B.1: Degenerate element counts and repairing efficacy

atoms shifts with respect to another sheet. This type of slip propagates quickly through the solid, but degenerate elements block the propagation and create non-physical dislocation buildups.

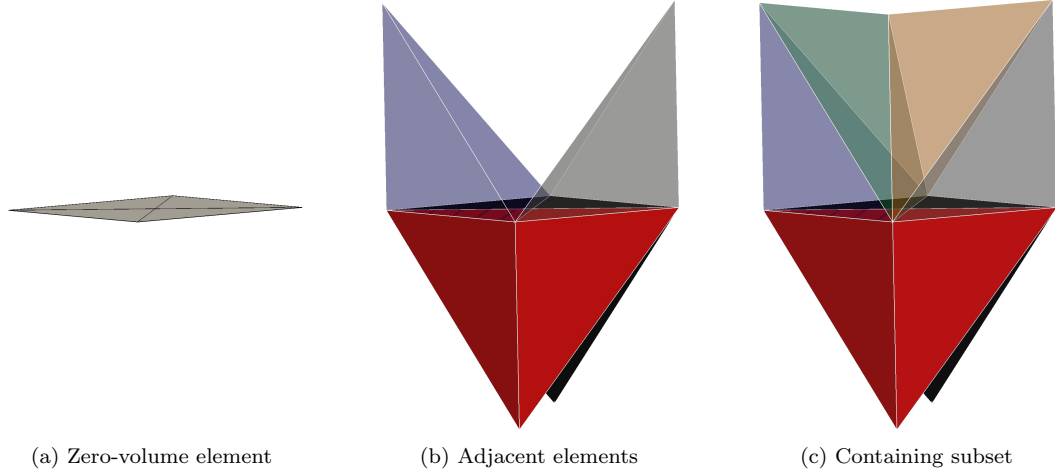


Figure B.4: Geometry that cannot be repaired.

B.3 Repairing algorithm and efficacy

The core strategy for repairing the mesh of its degenerate zero-volume elements is to identify subsets of the mesh that contain each degenerate element. Once a subset is identified, the internal topology of the subset is modified (with a standard 3-2 edge flip [Shewchuk, 2002]) in a way that eliminates the zero-volume tetrahedron while maintaining external shape and topology. The maintenance of

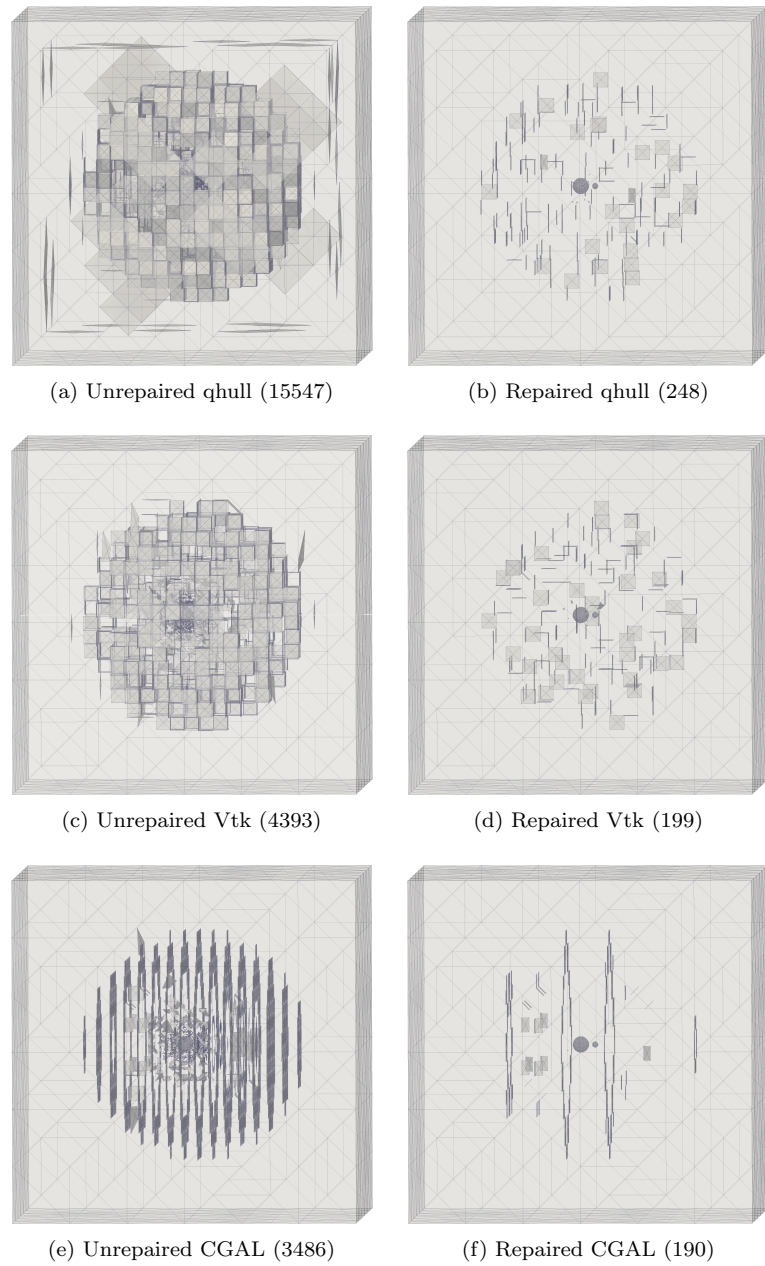


Figure B.5: Repairing the structured mesh from a scenario with two voids.

external topology limits what geometries can be corrected. For example, the three geometries of Figure B.1 can be successfully repaired: the first row shows an isolated zero-volume degenerate element, the second row shows an isolated element of small but nonzero volume, and the last row shows two adjacent degenerate tetrahedra. However, the geometry in Figure B.4 cannot be repaired; the subset of elements containing the degenerate tetrahedron can be identified and is shown, but there is no way to reconnect the elements in a way that maintains external shape and topology. This

specific irreparable geometry is a common defect found in meshes that use spherical coarsening and accounts for the vast majority of the remaining (unrepaired) degenerate elements in Figure B.5.

At a high level, an iteration of the repairing algorithm uses a given element condition number to identify “degenerate” elements to attempt to repair, then attempts to repair all of that set of degenerate elements. As enumerated in Listing B.1, the repairing process is performed in stages where the degenerate element condition number is reduced from a starting value (for example, 20) to a final value (for example, 5) over the course of multiple passes.

Listing B.1: Mesh repair algorithm

```

for each decreasing degenerate element condition number
  make a list of all degenerate elements
  for each degenerate element
    if it has exactly one adjacent degenerate element
      attempt special case of two adjacent elements
    if it has no adjacent degenerate elements
      if the correct geometry (a pyramid on one side) is be found
        identify candidate correction
      if the maximum condition number of the new simplices is less
        than the condition numbers of the original simplices
        modify two simplices, remove degenerate simplex

```

Table B.1 shows the number of degenerate elements produced by the various meshers for four illustrative scenarios, as well as the repair success rate. Qhull creates the largest number of zero-volume elements, even though its “joggling” option is not being used. Vtk produces fewer than qhull, but because manual “joggling” of the input points is necessary for solutions to be found, many zero-volume elements result. CGAL and tetgen perform almost identically, even sharing the vast majority (> 99%) of their defects. Matlab and Mathematica create similar numbers of degenerate elements, but the defects are not shared; they were included in this study for comparison and are not directly usable by the code because of the complication of interfacing with them, the extra software dependencies they would introduce, and the limitation in the size of meshes they can realistically create.

In general, across the tens of different scenarios on which the mesh repair algorithm has been used, it seems to eliminate over 98% of the defects for geometries of structured, random, or perturbed meshes (as illustrated by the first three scenarios in Table B.1). Scenarios (such as the TwoVoids scenario in the table) with spherical coarsening exhibit a specific defect (see Figure B.4) that is not repairable by the algorithm, and defect repair rates drop to around 95%.

Specifically, Figures B.2 and B.3 show the degenerate elements before and after repair for the

cases of sharp and diffuse 3D mesh interfaces. In the scenario with sharp mesh interfaces (Figure B.2), CGAL/tetgen produces defects in the region around the atomistic part of the domain, while Vtk and Qhull produce many more degenerate elements throughout the domain. However, the mesh repair algorithm successfully removes all produced defects, regardless of the original mesher. In the scenario with diffuse mesh interfaces (Figure B.3), repairing removes the vast majority but not quite all (2-3 are left) of the produced defects, regardless of the original mesher.

Unlike for the previous interface scenarios, the mesh used for the study of elastic moduli (see Section 3.3.2) is generated with random repatom positions and has no underlying structure. This randomness reduces the performance gap between the best- and worst-performing meshers, and all meshers produce hundreds of degenerate tetrahedra (see Figure B.6). Repairing successfully removes more than 98% of the defects, and around 10 remain.

Figure B.5 shows a scenario in which there are two voids of different sizes, around each of which the mesh is spherically coarsened (see section 4.3.1.1). In many scenarios, the more poorly-performing meshers create similar defects but produce more of them. For this two voids scenario, CGAL, tetgen, Matlab, and Mathematica all produce very similar defect patterns and only differ by around a hundred defects but qhull and vtk produce qualitatively different mesh defect patterns. This scenario is clearly a difficult one for qhull’s algorithm, as it produces four times more defects than the other meshers. The mesh repair algorithm repairs most of the defects, but cannot repair the specific defect shown in Figure B.4, and around 200 of these defects are left in the mesh.

B.4 Rounding point coordinates

The degenerate tetrahedra do not represent *errors*, per se, on the part of the meshers. When input points are co-spherical, floating point approximation and tolerance become significant in the evaluation of geometric predicates. Given the input point locations and their approximate floating-point representations, the correct Delaunay triangulation must include those elements.

One way to reduce the number of produced degenerate tetrahedra is to improve the evaluation of the geometric predicates by “snapping” input point coordinates to some given precision. That is, two points may conceptually have the same x coordinate, but their x values are 6.54321 and 6.54320. By rounding coordinates to a precision of 1×10^{-2} , the mesher can recognize that the points are on the same plane and perhaps not generate a zero-volume element.

Figure B.7 shows the reduction in zero-volume tetrahedra versus the precision used in the rounding process for the “Diffuse Interfaces” (Figure B.3), “Elastic Constants” (Figure B.6), and “Two Voids” (Figure B.5) scenarios. The number of degenerate elements converges to the un-rounded number as the precision is reduced, and the optimal precision for all three scenarios is 1×10^{-3} .

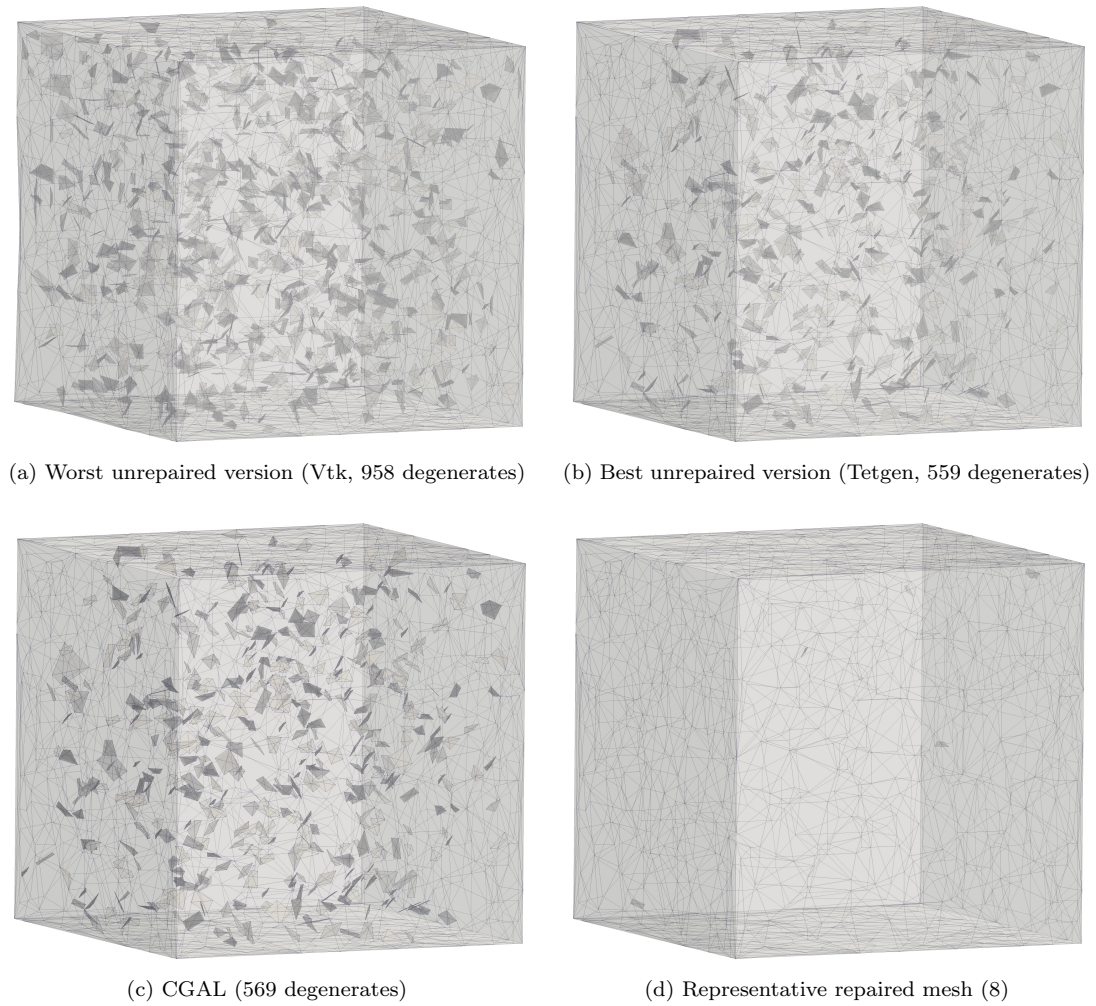


Figure B.6: Repairing the random mesh from the Elastic Constants scenario [3.3.2](#).

B.5 Chosen meshing strategy

The performance numbers and examples of Section [B.3](#) are representative of the general performance trends observed throughout the body of simulations that have been performed. CGAL and Tetgen create almost identical meshes, while Qhull and Vtk generally produce many more degenerate elements. Both Tetgen and CGAL have mechanisms for computing *exact predicates*, the evaluation of which is crucial for calculating the `incircle` test. Tetgen is a direct derivative of Shewchuk’s triangle code [[Shewchuk, 1996](#)] and uses its exact geometric predicates [[Shewchuk, 1997](#)] described [here](#). CGAL provides exact predicates and constructions through its `Exact_predicates_exact_constructions_kernel` and `Exact_predicates_inexact_constructions_kernel` geometry kernels which implement the “exact computation paradigm” [[Yap and Dube, 1995](#)]. Because use of the kernel with inexact constructions does not change the degenerate elements nor the resultant mesh

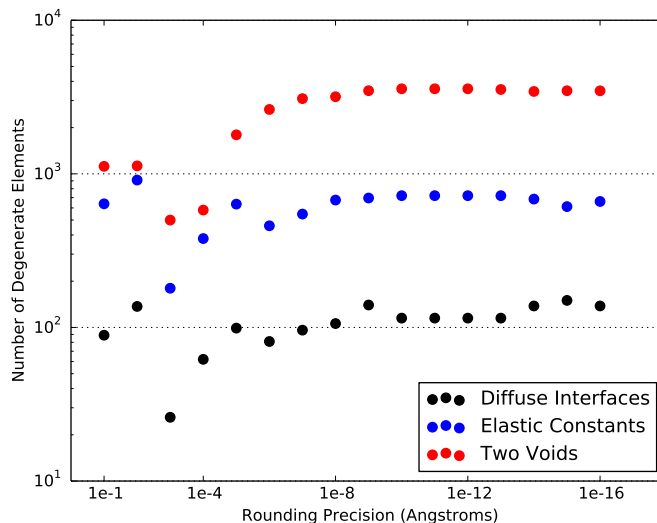


Figure B.7: Reduction in zero-volume tetrahedra by rounding point coordinates.

and is significantly faster, it is used in the current code.

The combination of Triangle/Tetgen performs almost identically to CGAL, but is not used because of software difficulties that required both to be modified. In particular, because Tetgen is a direct derivative of Triangle, it is practically quite difficult to have both Tetgen and Triangle in the same software project because they define symbols with the same name in the global namespace. Tetgen is not set up to perform many consecutive meshes and required modification to better handle its error and edges cases, because meshing is performed millions of times for the generation of optimal sampling atoms.

CGAL makes meshes that are almost identical to those of Tetgen, it completes in about half the time of Tetgen, supports 2D and 3D meshing, is actively maintained by an entire team of developers, and does not require any modification. For all of these reasons, CGAL is currently the recommended mesher.

Figure B.8 shows the effect of rounding, repairing, and their combination. Rounding alone decreases the number of degenerate elements by sometimes as much as an order of magnitude or more, while repairing alone usually reduces the number by several orders of magnitude. The combination of rounding and repairing CGAL meshes successfully *removes almost all defects, leaving no more than 3 in even the hardest meshes*.

Though there are some rare cases in which the process of rounding point coordinates increases the number of degenerate tetrahedra, the combination of rounding and repairing produces the fewest number of zero-volume elements. Because of this, CGAL is used for all 3D meshing, and rounding

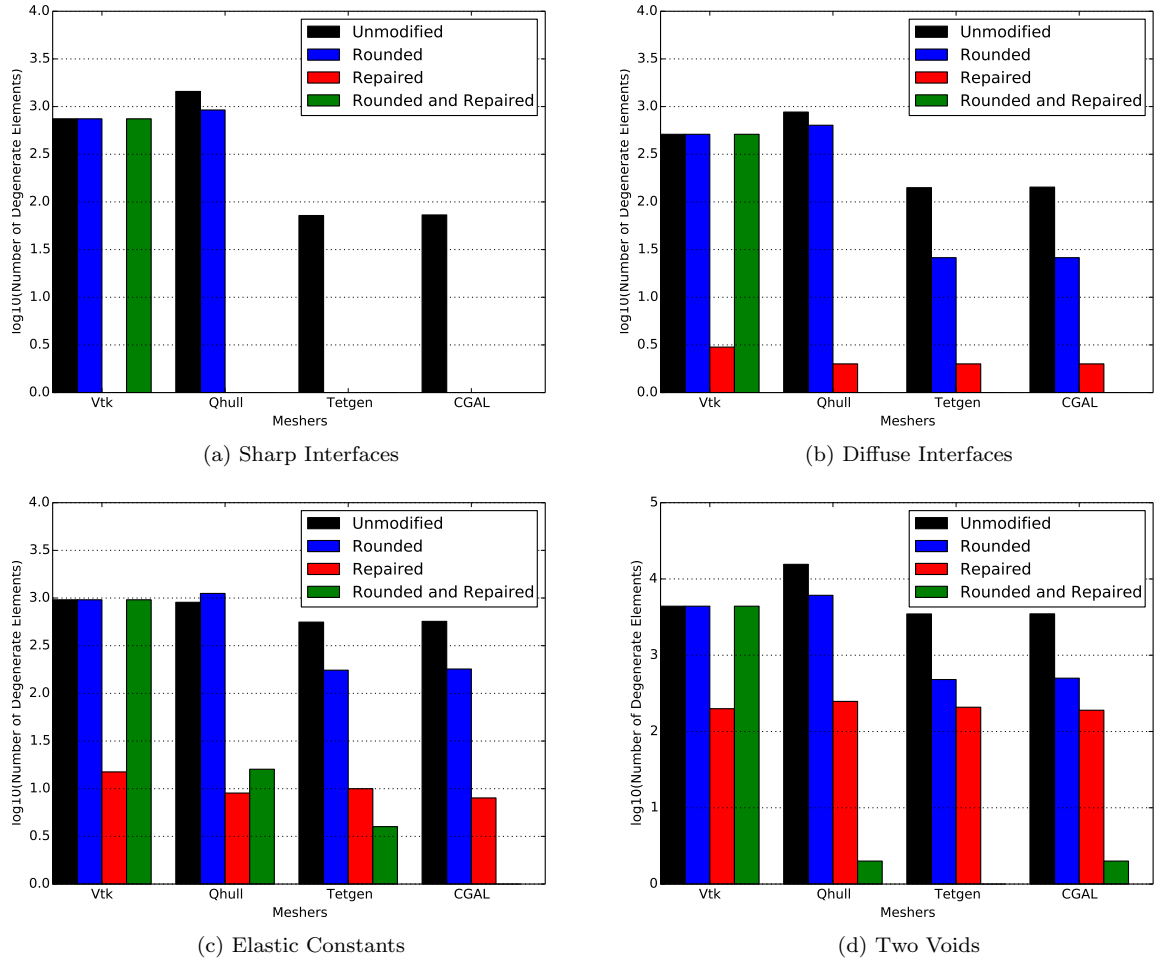
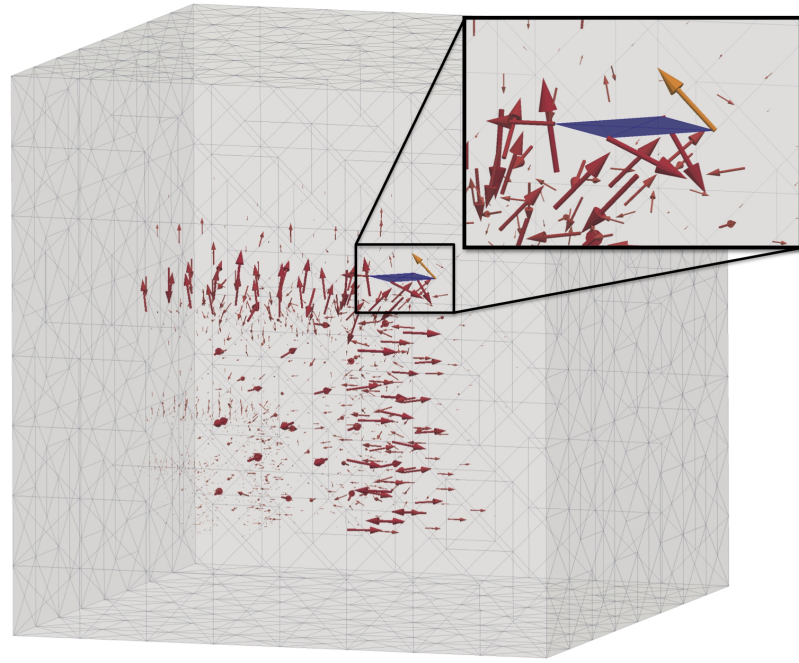


Figure B.8: Performance of repairing and rounding for various scenarios.

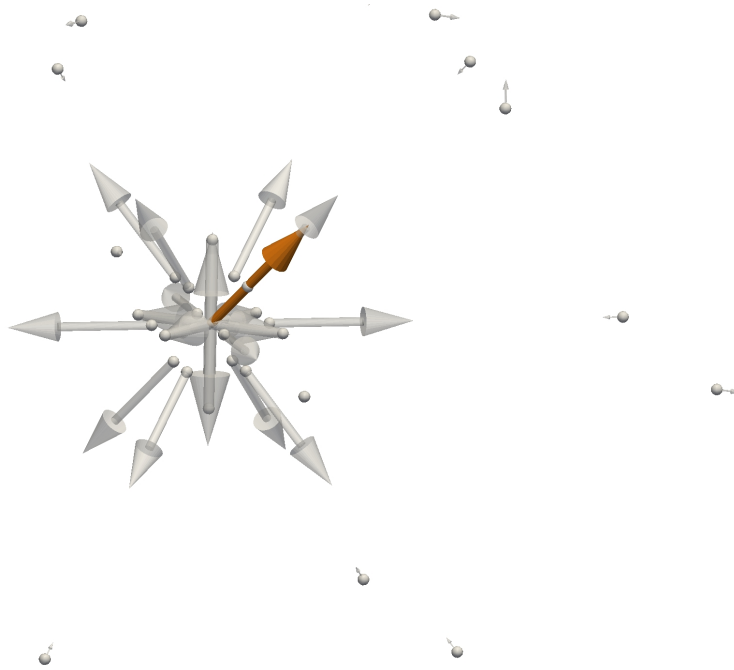
and repairing is automatically performed on all meshes of *repatom positions*. Meshes performed in optimal sampling atom population (described in Section 4.5.2) to determine the total volume of the convex hulls do not use either rounding or repairing, because correct internal topology is not necessary to determine the total volume of the convex hulls.

B.6 Ghost forces from mesh defects

Zero-volume elements not only cause difficulties with optimal sampling atom population and mesh refinement, but also directly create ghost forces. Unlike the ghost forces created by incorrect energy integration from summation rules, mesh defects cause ghost forces which are not inherent in the formulation would not exist in a mesh without defects. For example, Figure 4.3 shows that the ghost forces from mesh defects can be of the same order of, or even significantly larger than, the forces caused by incorrect energy summation. For the worst case scenarios, these errors can reduce



(a) Isolated degenerate element



(b) Ghost force (orange) and contributing forces (white)

Figure B.9: Ghost force arising from mesh defect.

the accuracy of even the best summation rules down to the level of the worst. The ghost forces arise from two sources: finite jumps in shape function values from infinitesimal perturbations in space as

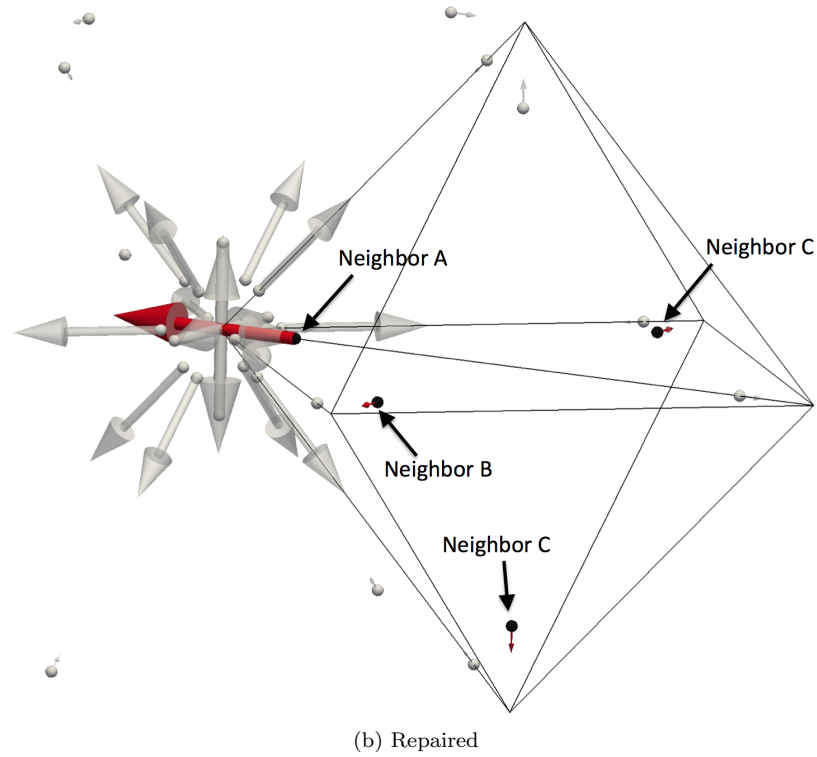
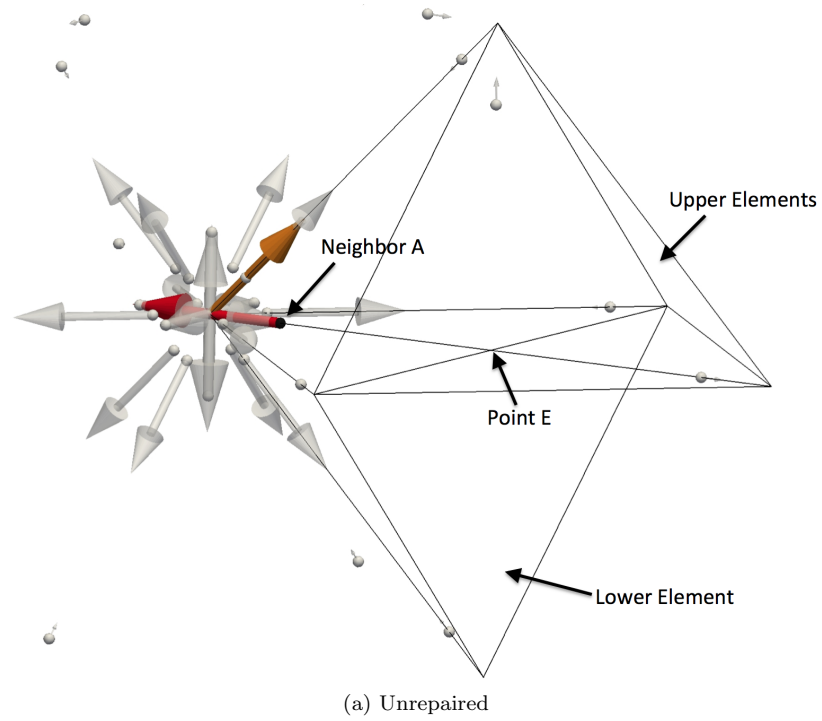


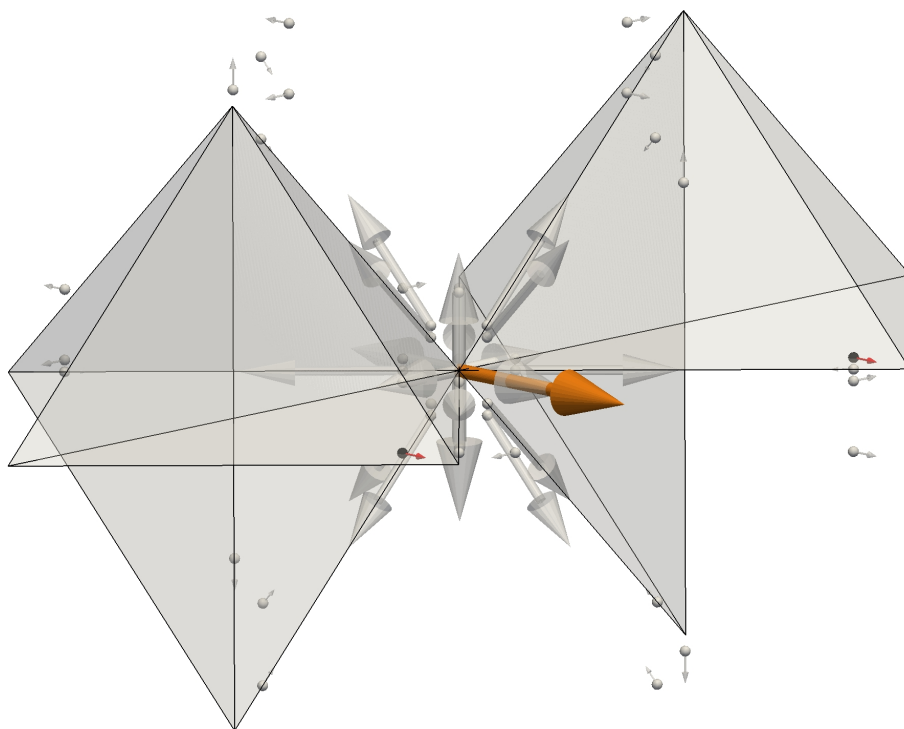
Figure B.10: Repairing removes the ghost force (orange) arising from the mesh defect. Red forces are different between the unrepaired and repaired configurations.

well as missing neighbors.

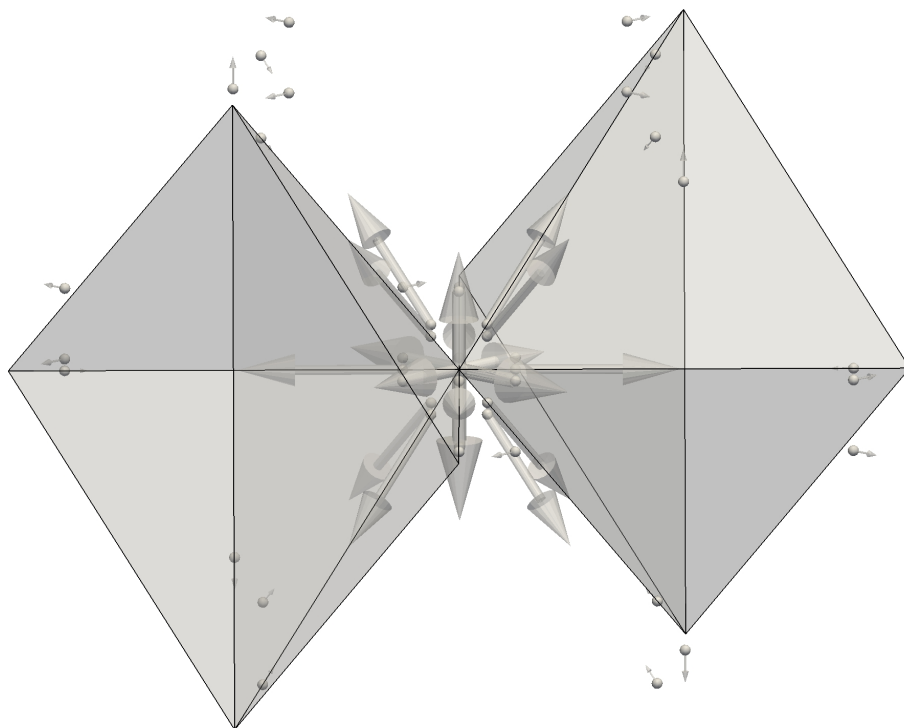
The total force on a given repatom comes from two sources shown in Equation 4.1: nonzero shape function values associated with sampling atom neighbors ($N_k(\mathbf{X}_j)$) and nonzero shape function values associated with sampling atoms ($N_k(\mathbf{X}_\alpha)$). For every sampling atom neighbor which has a nonzero shape function value associated to a given repatom, the atomic force is calculated between the neighbor and the sampling atom, it's multiplied by the shape function value, and added to the repatom's force.

For example, Figure B.9 shows the ghost force that results from leaving a single degenerate element unrepaired. Panel (a) shows the zero-volume element in relation to the rest of the sample and the ghost forces that result from the summation rule. The incorrect ghost force is colored orange, but is drawn on the same scale as the “legitimate” ghost forces in red. Panel (b) shows the forces on each sampling atom neighbor that sum together to create the orange ghost force. Figure B.9 shows the unrepaired and repaired elements; the red forces in the panels highlight forces which are different between the two states.

There are two categories of problems fixed by repairing: missing neighbors and incorrect shape function values. With regards to missing neighbors, due to the mismatch in the elements in the unrepaired state, three neighbor contributions are missing (neighbors B , C , and D) because the repatom's shape function value is zero at their locations. With regards to incorrect shape function values, shape function values are computed by first finding the containing simplex of every neighbor location and then computing the values. When a point (such as neighbor A) is directly on an edge, noise in floating point calculations affects which of the incident elements is determined to contain that neighbor. Shape function value discontinuities arise from mesh defects because the shape function value is different depending on whether the neighbor position is determined to be located within the lower element or either upper element. Consider, for example, a theoretical neighbor on point E . If the shape function value is computed with an upper element, it is 0.5, whereas it will be 0.0 if the lower element is used. In this particular example, neighbor A is determined to be in the lower element for the unrepaired configuration, and as such the force contribution from that neighbor increases when the elements are repaired. Figure B.11 shows a similar situation where the ghost force is formed by two degenerate elements simply because of incorrect neighbors that are included in the unrepaired state.



(a) Unrepaired



(b) Repaired

Figure B.11: Repairing removes the ghost force (orange) arising from the mesh defect. Red forces are different between the unrepaired and repaired configurations.

Appendix C

Parallelization

Many techniques can be implemented to reduce the runtime of the implementation when it is run on a single computer (node). The tools and technologies that have been used in the current implementation are cache-aware data structures and algorithms (to minimize time spent waiting on cache misses), malloc-avoiding data structures (to avoid the costly process of allocating memory), SIMD vectorization (to use vector registers on modern processors), and threading (to use all of the computational cores of an individual processor). Though these tools are critical for a highly-performing realization of the method, they are ultimately limited to only achieving the throughput accessible by a single node. To tackle problems of engineering or scientific significance, a realization must use distributed-memory parallelism so that it can be run on common large-scale computer clusters.

C.1 Background

At a high level, using distributed-memory parallelism means writing a program, running multiple copies of that program on different computers, and having the multiple copies of the program work together to either solve a problem faster than they could individually solve the problem, or to solve a larger problem in a similar amount of time. As described in [Section 4.4](#), there are two methods to parallelize any given problem for distributed-memory machines, and the fundamental difference is whether each instance of the program “sees the big picture” or not. We will consider molecular dynamics as a pertinent example.

Listing C.1: Basic MD structure

```

generate atomic positions
for each timestep
  fill atomicForces with zeros
  for each atom  $a$ 
    find neighbors of  $a$ 
    for each neighbor  $n$ 
      force = calculate force between atoms  $a$  and  $n$ 
      atomicForces[ $a$ ] += force
  apply external forces and boundary conditions to all atoms
  update atomic velocities and positions

```

In MD, the entire state of the simulation is the positions (and associated data) of all of the atoms. A typical serial MD solver has a structure similar to Listing C.1, where some (essential) details (such as the use of Verlet neighbor lists) have been omitted.

Listing C.2: “Big Picture” parallel MD structure

```

generate atomic positions
for each timestep
  fill atomicForces with zeros
  for each atom  $a$  that this instance is in charge of
    find neighbors of  $a$ 
    for each neighbor  $n$ 
      force = calculate force between atoms  $a$  and  $n$ 
      atomicForces[ $a$ ] += force
  combine atomicForces across all instances
  apply external forces and boundary conditions to all atoms
  update atomic velocities and positions

```

The easiest and fastest way of introducing distributed-memory parallelism to the MD solver shown in Listing C.1 is to use the “Big Picture” paradigm, where every instance of the program has the entire state of the simulation. On each timestep, each instance updates a portion of the total state, and all instances share their updated portions with all other instances. This type of structure is shown in Listing C.2.

The parallelization strategy shown in Listing C.2 is quick, easy, and it scales moderately well. The only real limitation of this “big picture” approach is the size of problem it can be used to solve,

precisely because every instance of the program must be able to store the entire program state. Even if an MD solver had no other data structures than simply the atomic positions, velocities and forces, the largest number of atoms that can fit on compute node with two gigabytes (Gb) of RAM is around 20 million (in reality, the number is lower because of other necessary data structures). This is a severe limitation: with “big picture” parallelism no simulation with more than 20 million atoms can be performed with even the largest supercomputers because the entire state of the simulation cannot fit on any given node.

Listing C.3: “Need to Know” parallel MD structure, bolded sections denote differences between parallel and serial algorithms

```

generate atomic positions of the subset that this instance is in charge of
for each timestep
    send information to instances that need things from us
    receive information we need from other instances
    fill atomicForces with zeros
    for each atom a on this instance
        find neighbors of a
        for each neighbor n
            force = calculate force between atoms a and n
            atomicForces[a] += force
    apply external forces and boundary conditions to all atoms on this instance
    update atomic velocities and positions of all atoms on this instance

```

In order to perform large-scale simulations and fully utilize large supercomputers, each instance of the program cannot hold the “big picture” but only the subdomain that it “needs to know”. For typical domain decomposition problems, each instance has a subset of the physical domain that it is “in charge of”. However, for each instance to update the simulation state in this subset, the instance must have some information from adjacent or adjoining regions. This extra information is typically referred to as “ghost”, “shadow”, or “halo” information. The amount of extra information required is a function of how local the fundamental mechanical interactions are. Listing C.3 shows a typical “need to know” style parallel MD solver.

When each instance only has what it needs to know, simulations of arbitrary size can be performed as long as one has a sufficiently large computer. Because the communication performed by each instance scales with the number of neighbors of each instance and not with the total size of the simulation, “need to know” implementations can scale extremely well (in the *weak* sense, where N instances can perform a simulation of size $N * S$ in the same amount of time as one instance performs a simulation of size S).

However, writing a distributed-memory-parallel program in the “need to know” paradigm is extremely hard, precisely because no one instance of the program sees the entire picture. Many operations may require the total state, such as calculating total equilibration errors, finding the total power for the FIRE update (Section 4.2.1), or load balancing. These operations require reformulation or other strategies to eliminate the need to store the entire simulation state on a single computer at any given time. Global communication may be performed for some of these operations, but global communication ultimately limits scaling and should be avoided.

C.2 Formulating QC for distributed memory parallelism

The novel QC formulation of this thesis combines aspects and ideas from both MD and mesh-based techniques such as finite elements. In this work, the state of a QC simulation contains undeformed repatom positions, deformed repatom positions, a mesh connecting the undeformed repatom positions, and sampling atoms. In order to parallelize QC under the “big picture” paradigm (where all instances of the program contain the entire simulation state), any given instance of the program must store all of these data structures for the entire simulation.

A rough estimate on the limit of the simulation size possible using big picture parallelism is motivational for why the current realization does not use it. In practice, the number of elements in a mesh scales roughly as six times the number of repatoms. The number of sampling atoms depends on the summation rule, but can scale roughly as four times the number of repatoms (for the more-accurate summation rules of multi-point quadrature or second-order optimal). Given the sizes of the repatoms, elements, and sampling atoms, if there were no other data structures in the simulation then the largest QC simulation that can fit on a compute node with two Gb of RAM would have around two million repatoms. For compute nodes with large amounts of RAM (such as 48 Gb, which is large in contemporary practice), simulations of a few tens of millions of repatoms are possible. Performing simulations of tens of millions of repatoms represents a significant advancement in the size of possible QC simulations, but is smaller than the simulations ultimately targeted by this work. Because of this, the current realization uses the “need to know” paradigm, which significantly complicates the implementation.

C.2.1 State decomposition

The state of an MD simulation is simple, compact, easily divisible, and lends itself naturally to distributed-memory parallelization. Because of the finite radius of interaction between any two atoms in the simulation, communication dependencies and relationships between instances of the program are easy to determine. The QC formulation described in this work is substantially harder to parallelize because the state is not easily divisible and communication dependencies are much

more difficult to determine.

The state of a QC simulation is the combination of repatoms, sampling atoms, and a mesh. Therefore, the problem decomposition can be performed on any one of those data structures: either each process owns a nonoverlapping subset of the repatoms, or a nonoverlapping subset of the sampling atoms, or a nonoverlapping subset of the mesh.

Repatom decomposition In a repatom-based decomposition where every instance owns a nonoverlapping subset of the repatoms, each instance is in charge of calculating the force on, and updating the position of, all owned repatoms. This means that every instance of the program must have all of the sampling atoms which either depend on, or have a neighbor which depends on, every owned repatom. In order to correctly calculate the forces from each of these necessary sampling atoms, each instance of the program must also have all of the mesh required to interpolate the positions of every neighbor of every one of these necessary sampling atoms. Note that duplicate copies of sampling atoms will exist across the simulation because any given sampling atom may contribute to repatoms owned by several different instances of the program.

Sampling atom decomposition In a sampling-atom-based decomposition where every instance owns a nonoverlapping subset of the sampling atoms, each instance is in charge of calculating the forces associated with all owned sampling atoms. These forces make contributions to repatoms in a subset of the domain. The total global force on each repatom must be collected from all contributing instances of the program, after which that repatom's position must be updated by the owner of the repatom. Therefore, this sampling atom decomposition also requires a decomposition of the repatoms. Instances of the program communicate force contributions to the owner of that particular repatom, after which the owner collects the forces, updates the position, and sends the updated position to all instances which need it.

The repatom-only decomposition duplicates work because the forces from any single sampling atom may be computed many times (once per instance that needs it). Also, because of the mesh required to support all of those contributing sampling atoms, more communication is required than for the sampling atom decomposition. Therefore, the sampling atom decomposition is used in the current realization of the method.

Listing C.4: Serial initialization of a QC simulation

```

populate undeformed repatom positions using the given scenario
use an external mesher to create a global mesh of the undeformed repatom positions
calculate repatom weights
use the specified summation rule to create sampling atoms in the entire mesh
calculate sampling atom weights

```

C.3 Parallel initialization of a QC simulation

The serial initialization of a QC simulation is straight-forward (see Listing C.4) and its main steps are to create a mesh from undeformed repatom positions, populate sampling atoms, and calculate weights. Performing a distributed-memory parallel initialization is more complicated and requires partitioning of data structures (see Listing C.5).

Listing C.5: Parallel initialization of a QC simulation

```

populate undeformed repatom positions using the given scenario.
use an external mesher to create a global mesh of the undeformed repatom positions.
use an external partitioner to partition the global mesh such that each element
and each repatom has a unique owner.
forget about all of the state except the owned elements.
calculate repatom weights.
use the specified summation rule to create sampling atoms in owned elements.
remove duplicate sampling atoms across instances.
calculate sampling atom weights.
obtain all of the mesh required to support the neighbors of all owned sampling atoms.
form the communication map.

```

One of the core steps of the initialization process is to create a mesh of the undeformed repatom positions. There does not currently exist software to perform large-scale parallel Delaunay tetrahedralization. As such, the general strategy is to create a global mesh, partition the mesh, and proceed with initialization. It should be noted that this strategy requires that an instance of the program store the global mesh, which limits the total possible simulation size and is contrary to the idea of “Need to Know” parallelism described above. However, this step is only performed at the very beginning of a simulation, when the mesh is the coarsest. The size of the simulation grows with mesh refinement, which does not necessarily require the formulation of an entire global mesh.

Once the simulation starts and each instance of the program has the subdomain it is in charge of, the instances never again need to store the global mesh.

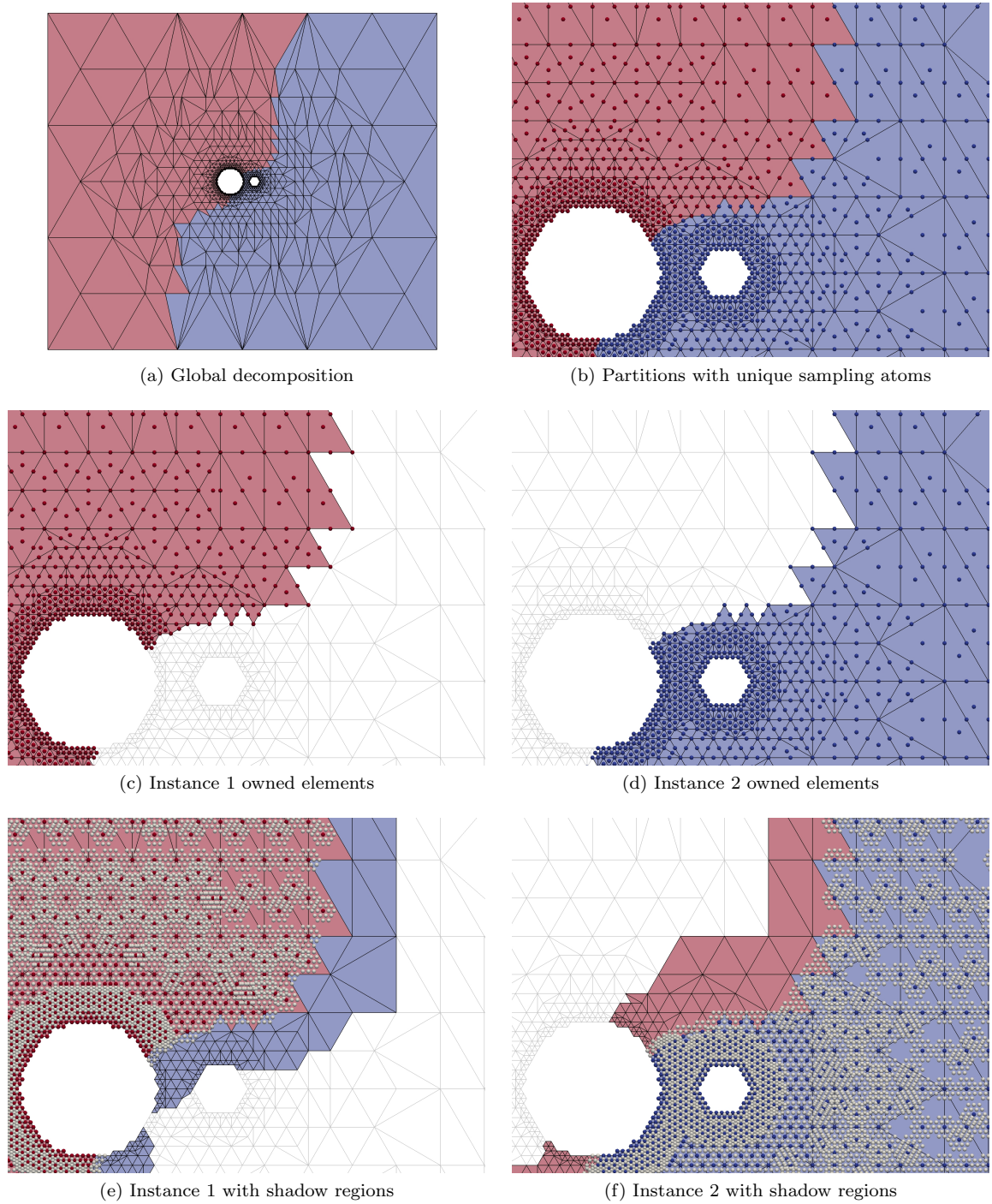


Figure C.1: The process of parallel initialization. Elements and sampling atoms are colored by owner (red or blue); neighbor atoms are colored white.

Once the mesh is partitioned, all instances of the program “forget about” the global mesh and restrict all of their data structures to solely the portion of the mesh that the instance is in charge of. The next step is to populate sampling atoms, which each instance does for all owned mesh elements, as shown (for the second-order optimal summation rule) in panels (c) and (d) of Figure C.1. Some sampling atoms are duplicated because they are created by two or more instances, which must be corrected because the parallel decomposition requires that every sampling atom exist on only one instance (panel (b) of Figure C.1 shows only the unique sampling atoms). Lastly, since each sampling atom only exists on a single instance of the program, that owning instance of the program must have all of the mesh required to interpolate the positions of all of that sampling atom’s neighbors.

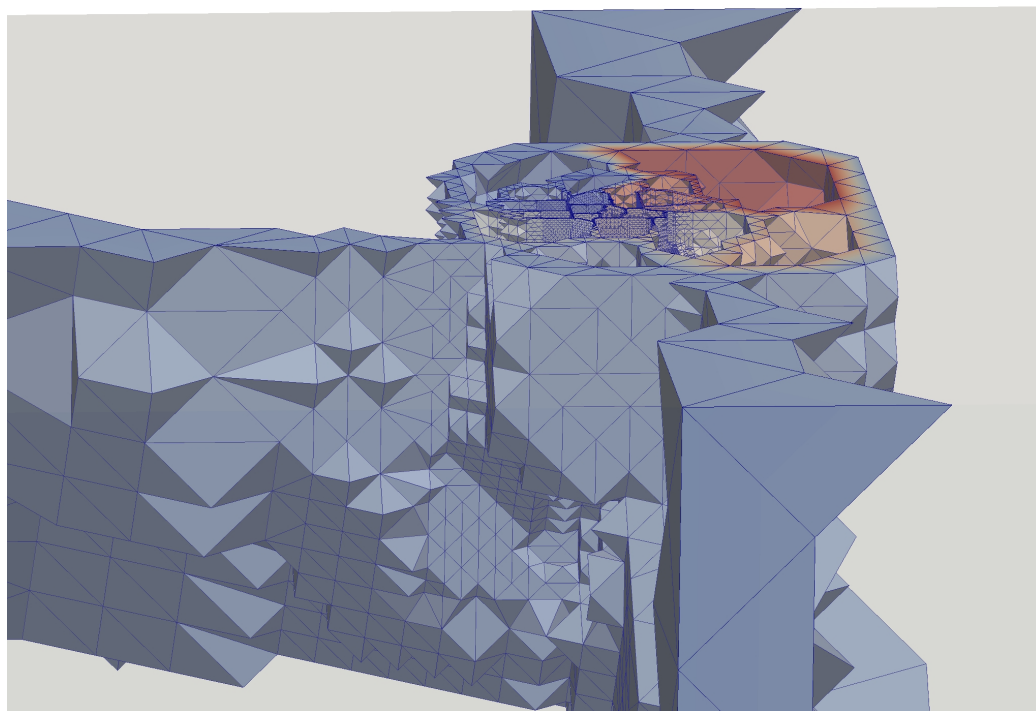
After sampling atom creation, the instances do not have enough supporting mesh to create neighborhoods for the sampling atoms near the exterior of the subdomains, and as such communication is required. This extra unowned mesh, which each instance needs to have to interpolate sampling atom neighbor positions, is referred to as a *shadow region*.

Panels (e) and (f) of Figure C.1 show the sampling atom neighbors and shadow regions on each instance. The shadow region is significantly larger region than that which is strictly required to support the sampling atom neighbors. As discussed in Section C.4, shadow regions are grown in large increments to reduce the number of times that program instances need to ask their neighbors for more mesh.

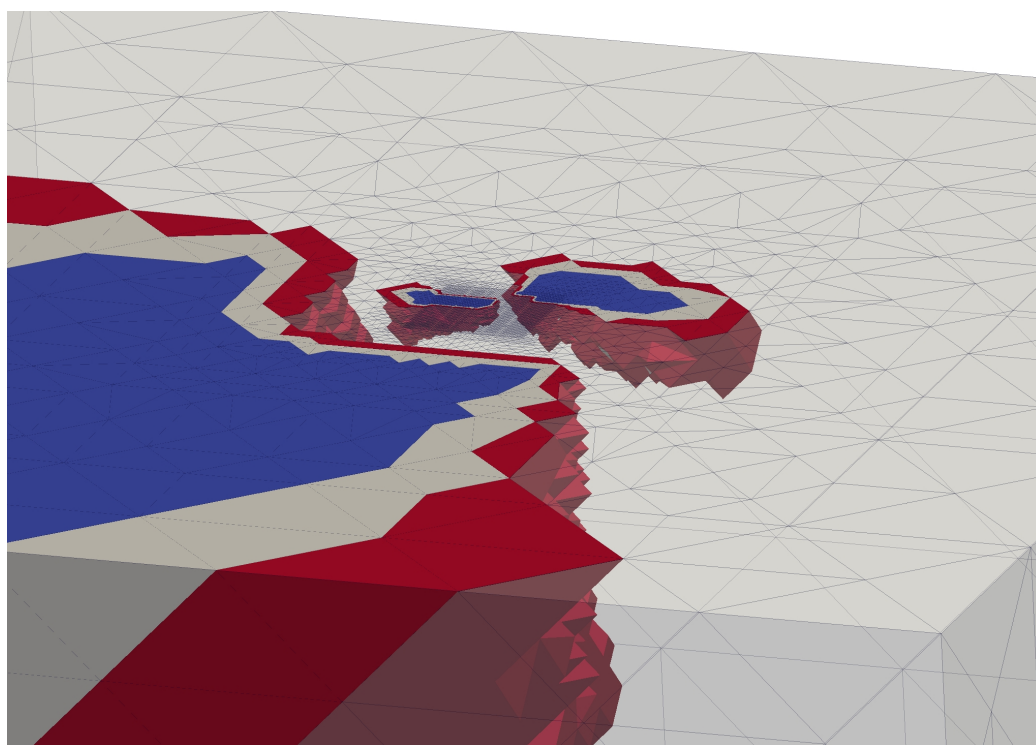
As another example of domains in parallel simulations, Figure C.2 shows the domain decomposition for a 3D nanoindentation example. In panel (a), the elements on the boundaries of instance subdomains are opaque and the remainder is semi-transparent. In panel (b), the elements owned by an instance are colored blue, shadowed elements are colored red, and shared elements are colored white.

Every repatom has a unique owning instance, but because of the mesh partitioning the total force on a given repatom may be distributed into contributions that are found on several instances. Because of this, once force contributions are calculated, each instance must send the forces on unowned repatoms to the owner of the repatom. Once the forces are collected, each instance can update the positions of all owned repatoms. However, any instance of the program that has a copy of that repatom in its mesh must now have the updated position. Therefore, after updating positions, each instance sends the updated positions to all other instances that need those particular positions. This interaction between parallelization, the force calculator, and the solver is shown in Listing 4.5.

Eliminating communication cost The cost of communicating repatoms can be significant in many distributed-memory programs. To reduce the cost of communication, three techniques are used. First, nonblocking communication is used to reduce coordination and synchronization time. Secondly, communication happens concurrently with calculation. That is, each instance calculates



(a) Subdomain boundaries



(b) Owned (blue), shared (white), and shadowed (red) elements

Figure C.2: Nanoindentation domain decomposition example.

updated values that it will have to send to others, starts communication for those values, and then calculates the remainder of the values *while* the communication is happening. Lastly, communication is performed in a special communication thread to enable communication in the background. By using these strategies, the cost of communication can be greatly reduced or completely eliminated, though this elimination exacerbates any existing problems with load balancing.

C.4 Adaptive neighborhoods in parallel simulations

The operating assumption behind the parallel formulation of the algorithm is that, once the simulation initialization has completed, it is not feasible to have the entire state on any instance of the program. Anytime an instance of the program either needs more mesh than it currently has, or needs to change the mesh it has, nontrivial communication is required.

This design principle complicates the adaptive high-deformation neighborhoods algorithm described in Appendix A. Without considering parallelization, the neighborhoods algorithm is based on a key distance: the largest distance in the undeformed configuration to a neighbor in the deformed configuration, the `maxNeighborDistance`. When building the neighborhood, the algorithm will generate candidate neighbors at all lattice sites within `maxNeighborDistance` plus a buffer, a distance we will term the `requiredSupportRadius`. All candidate neighbors are mapped to the deformed configuration, those which are within the `deformedCutoffRadius` are kept as neighbors, and the `maxNeighborDistance` is updated. We note here that the `maxNeighborDistance` for any given sampling atom is not necessarily a monotonic function as the simulation progresses, because neighbors from different locations in the undeformed configuration can slip in and out of the `deformedCutoffRadius`.

Even in simulations where the number or undeformed positions of sampling atoms on each rank are not changing, as the neighborhoods deform and their `maxNeighborDistance` values increase, the balls around sampling atoms with radius `requiredSupportRadius` may extend beyond the mesh on that instance of the program. In this situation, the candidate neighbors that are generated may not be sufficient, and the solution will be incorrect. For the solution to be correct, the program must first detect when the ball extends beyond the available mesh and then ask for, receive, store, and add the communication logic for any mesh which lies within the ball but is owned by another instance.

The logic for determining if a ball extends beyond the available mesh is to perform a ball-mesh intersection, then determine if any element intersecting the ball has an incomplete adjacency (in 3D, each tetrahedron should have four adjacent elements that share the four faces). Elements with incomplete adjacency are on the boundary of the instance's mesh and may either be on the boundary of the entire simulation domain (in which case there is no more mesh to add) or on the boundary

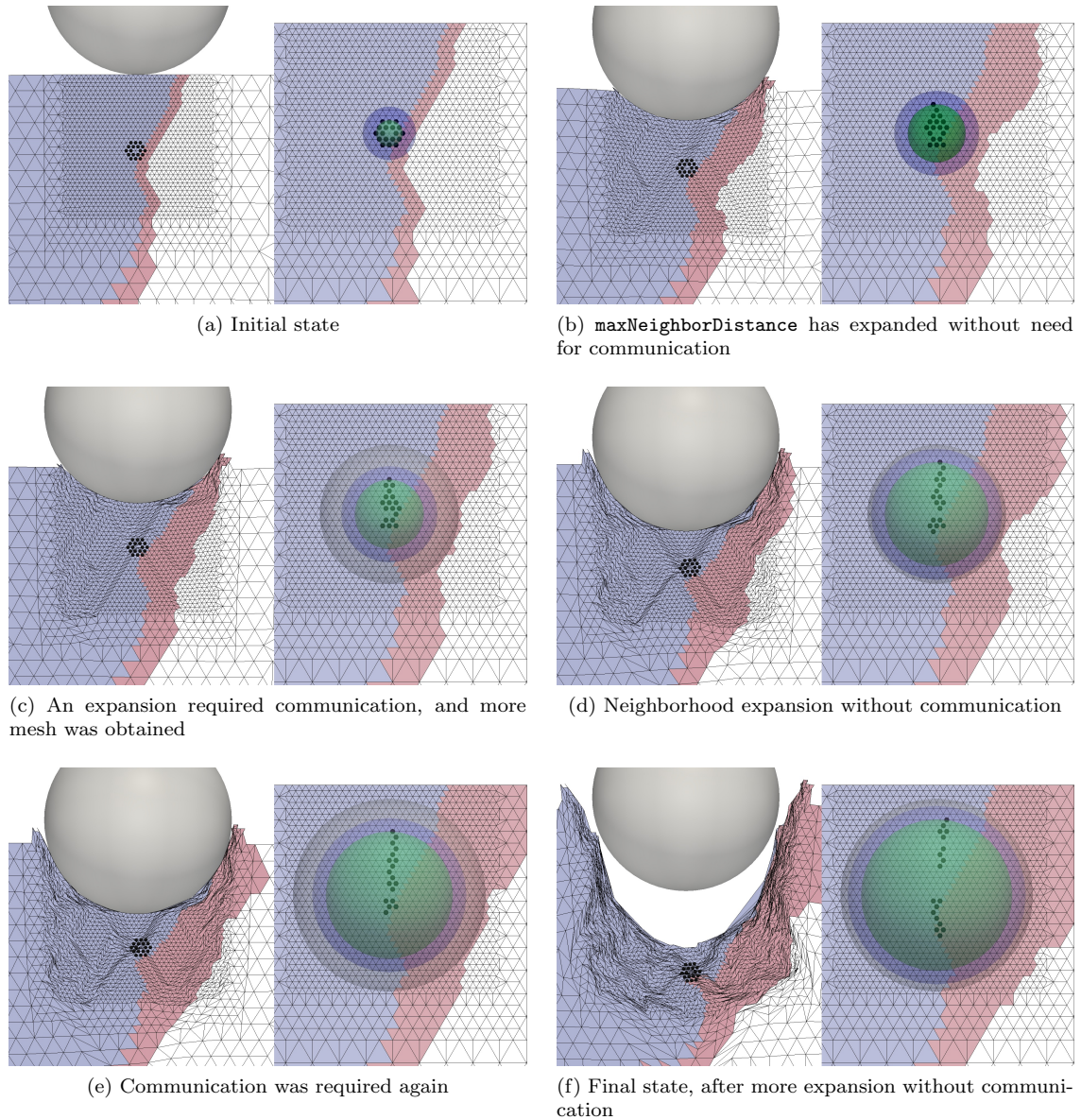


Figure C.3: Adaptive neighborhoods in parallel simulations. Each panel shows the undeformed (left) and deformed (right) configuration. Owned elements are blue, shadowed elements are red. Actual neighbors are black circles in both panels. Three semi-transparent spheres are drawn around the sampling atom: green (radius `maxNeighborDistance`), blue (radius `requiredSupportRadius`), and one gray (radius `meshSupportRadius`). A movie is available [here](#).

between the subdomains of two instances (in which case there is mesh to request and add).

The coordination and communication required to expand meshes is expensive and undesired. In order to reduce the amount of communication required by the adaptive neighborhoods, we introduce a `meshSupportRadius` for each sampling atom. This `meshSupportRadius` is the maximum distance of mesh support that the particular sampling atom is sure that it has access to. Whenever a sampling atom needs information at a given distance d , the code first checks if d is less than the

`meshSupportRadius` of that sampling atom, in which case no communication is required. If `d` is larger than that sampling atom's `meshSupportRadius`, then the aforementioned ball-mesh intersection is performed to determine if any intersecting elements are on the boundary of the program instance's subdomain. If no intersecting elements are on the boundary, then the instance knows that it has all of the mesh required to provide all neighbors within `d`, and the sampling atom's `meshSupportRadius` is set to `d`. If intersecting elements are on the boundary, then the instance must communicate with all other instances of the program to gather any mesh that overlaps with the ball of radius `d`. Once the communication happens, the instance knows that it has all of the mesh required to provide the neighbors within `d`, and the sampling atom's `meshSupportRadius` is set to `d`.

However, this approach (as described) may require frequent communication if a sampling atom's `d` is steadily increased, in which case communication is performed with each increase. To reduce the number of times that instances have to ask for mesh, we introduce a new parameter, the `meshExpansionDistance`. Once an instance determines that it needs communication to obtain more mesh, it does not ask for simply a distance of `d` around a particular sampling atom, but instead `d` plus `meshExpansionDistance`. This obtains more mesh than is strictly required, but no further communication is required for that sampling atom until `d` increases by `meshExpansionDistance`.

The process of obtaining mesh is illustrated in Figure C.3 (movie [here](#), which is an extreme nanoindentation scenario run on two instances of the program in which remeshing has been disabled in order to most clearly illustrate this process of obtaining new mesh. Panel (a) shows the initial state: the `maxNeighborDistance` is only the second shell and the `meshSupportRadius` is the same as the `requiredSupportRadius`. After some indentation (panel (b)), the program had to receive some mesh to correctly build neighborhoods of other sampling atoms and the red shadow region has grown. The `maxNeighborDistance` of the example sampling atom has grown, but when it grew there was already enough shadow region to support it, so no communication has been required for the example sampling atom. However, the state after the next loadstep is shown in panel (c), where the neighborhood had to grow in a way that required communication and the gray `meshSupportRadius` sphere is visible because it is now much larger than the `requiredSupportRadius` ball (in blue). Between the loadstep of panel (c) and the loadstep of panel (d), many neighborhood builds were performed but did not require communication because of the extra mesh that was obtained. In panel (d), the `requiredSupportRadius` has reached just less than the `meshSupportRadius`. On the next loadstep (shown in panel (e)), communication was required for a neighborhood build and the `meshSupportRadius` has again grown. More expansion is performed before the end of the simulation, but none required communication and the final state is shown in panel (f).

Bibliography

- F. F. Abraham, J. Q. Broughton, N. Bernstein, and E. Kaxiras. Spanning the continuum to quantum length scales in a dynamic simulation of brittle fracture. *EPL (Europhysics Letters)*, 44(6):783, 1998. URL <http://stacks.iop.org/0295-5075/44/i=6/a=783>.
- Ravi Agrawal, Bei Peng, Eleftherios E. Gdoutos, and Horacio D. Espinosa. Elasticity size effects in zno nanowires: a combined experimental-computational approach. *Nano Letters*, 8(11):3668–3674, 2008. doi: 10.1021/nl801724b. URL <http://dx.doi.org/10.1021/nl801724b>.
- J. S. Amelang, G. N. Venturini, and D. M. Kochmann. Summation rules for a fully-nonlocal energy-based quasicontinuum method. *J. Mech. Phys. Solids.*, 2015. doi: 10.1016/j.jmps.2015.03.007. URL <http://www.sciencedirect.com/science/article/pii/S0022509615000630#>. accepted for publication.
- M. Ariza, I. Romero, M. Ponga, and M. Ortiz. Hotqc simulation of nanovoid growth under tension in copper. *International Journal of Fracture*, 174:75–85, 2012. doi: 10.1007/s10704-011-9660-4. URL <http://dx.doi.org/10.1007/s10704-011-9660-4>.
- Steffen Arnold. Three-dimensional laser writing on the nanometer scale, 2013.
- A. Asthana, K. Momeni, A. Prasad, Y. K. Yap, and R. S. Yassar. In situ observation of size-scale effects on the mechanical properties of zno nanowires. *Nanotechnology*, 22(26):265712, 2011. URL <http://stacks.iop.org/0957-4484/22/i=26/a=265712>.
- Mehmet Balman. Tetrahedral mesh refinement in distributed environments, 2006.
- C.B. Barber, D.P. Dobkin, , and H.T. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. on Mathematical Software*, 22(4):469–483, Dec 1996.
- L. A. A. Beex, R. H. J. Peerlings, and M. G. D. Geers. A quasicontinuum methodology for multiscale analyses of discrete microstructural models. *Int. J. Num. Meth. Eng.*, 87(7):701–718, 2011. ISSN 1097-0207. doi: 10.1002/nme.3134. URL <http://dx.doi.org/10.1002/nme.3134>.

- LAA Beex, Pierre Kerfriden, Timon Rabczuk, and SPA Bordas. Quasicontinuum-based multiscale approaches for plate-like beam lattices experiencing in-plane and out-of-plane deformation. *Computer Methods in Applied Mechanics and Engineering*, 279:348–378, 2014a.
- L.A.A. Beex, R.H.J. Peerlings, and M.G.D. Geers. Central summation in the quasicontinuum method. *Journal of the Mechanics and Physics of Solids*, 70(0):242 – 261, 2014b. ISSN 0022-5096. doi: <http://dx.doi.org/10.1016/j.jmps.2014.05.019>. URL <http://www.sciencedirect.com/science/article/pii/S0022509614001100>.
- LAA Beex, RHJ Peerlings, and MGD Geers. A multiscale quasicontinuum method for dissipative lattice models and discrete networks. *Journal of the Mechanics and Physics of Solids*, 64:154–169, 2014c.
- T. Belytschko and S. P. Xiao. Coupling methods for continuum model with molecular model. *International Journal for Multiscale Computational Engineering*, 1(1):115–126, 2003.
- Marshall Bern and Paul Plassmann. Mesh generation. In *Handbook of Computational Geometry. Elsevier Science*, pages 291–332, 2000.
- J. Biener, A. M. Hodge, and A. V. Hamza. Deformation behavior of nanoporous metals. *Lawrence Livermore Laboratory Report*, UCRL-BOOK-218519, 2007.
- E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch. Structural relaxation made simple. *Phys. Rev. Lett.*, 97:170201, 2006.
- S. S. Brenner. Tensile strength of whiskers. *Journal of Applied Physics*, 27:1484–1491, 1956.
- S. S. Brenner. Plastic deformation of copper and silver whiskers. *Journal of Applied Physics*, 28: 1023–1026, 1957.
- Steffen Brinckmann, Dhiraj K. Mahajan, and Alexander Hartmaier. A scheme to combine molecular dynamics and dislocation dynamics. *Modelling and Simulation in Materials Science and Engineering*, 20(4):045001, 2012. URL <http://stacks.iop.org/0965-0393/20/i=4/a=045001>.
- J. Q. Broughton, F. F. Abraham, N. Bernstein, and E. Kaxiras. Concurrent coupling of length scales: Methodology and application. *Phys. Rev. B*, 60:2391–2403, 1999. doi: 10.1103/PhysRevB.60.2391. URL <http://link.aps.org/doi/10.1103/PhysRevB.60.2391>.
- Vasily V Bulatov, Luke L Hsiung, Meijie Tang, Athanasios Arsenlis, Maria C Bartelt, Wei Cai, Jeff N Florando, Masato Hiratani, Moon Rhee, Gregg Hommes, et al. Dislocation multi-junctions and strain hardening. *Nature*, 440(7088):1174–1178, 2006.

- M J Burek and J R Greer. Fabrication and microstructure control of nanoscale mechanical testing specimens via electron beam lithography and electroplating. *Nano letters*, 10(1):69–76, January 2010. ISSN 1530-6992. doi: 10.1021/nl902872w. URL <http://www.ncbi.nlm.nih.gov/pubmed/19961184>.
- C. Chen, Y. Shi, Y. Zhang, J. Zhu, and Y. Yan. Size dependence of young’s modulus in zno nanowires. *Phys. Rev. Lett.*, 96:075505, Feb 2006. doi: 10.1103/PhysRevLett.96.075505. URL <http://link.aps.org/doi/10.1103/PhysRevLett.96.075505>.
- Na Chen, Yiqun Xie, Feng Liu, Xiang Ye, and WangZhou Shi. A study of the size-dependent elastic properties of cdse nanowires. *Computational Materials Science*, 77(0):245 – 249, 2013. ISSN 0927-0256. doi: <http://dx.doi.org/10.1016/j.commatsci.2013.04.052>. URL <http://www.sciencedirect.com/science/article/pii/S0927025613002243>.
- Peter W. Chung. Computational method for atomistic homogenization of nanopatterned point defect structures. *International Journal for Numerical Methods in Engineering*, 60(4):833–859, 2004. doi: 10.1002/nme.989. URL <http://dx.doi.org/10.1002/nme.989>.
- K. L. Clarkson, K. Mehlhorn, , and R. Seidel. Four results on randomized incremental constructions. *Comp. Geom.: Theory and Applications*, pages 185–121, 1993.
- John D. Clayton and Peter W. Chung. An atomistic-to-continuum framework for nonlinear crystal mechanics based on asymptotic homogenization. *Journal of the Mechanics and Physics of Solids*, 54(8):1604–1639, 2006. doi: 10.1016/j.jmps.2006.02.004. URL <http://www.sciencedirect.com/science/article/pii/S0022509606000408>.
- H. L. De Cougny and M. S. Shephard. Parallel refinement and coarsening of tetrahedral meshes. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, 46:1101–1125, 1999.
- W. A. Curtin and R. E. Miller. Atomistic/continuum coupling in computational materials science. *Modelling and Simulation in Materials Science and Engineering*, 11(3):R33, 2003. URL <http://stacks.iop.org/0965-0393/11/i=3/a=201>.
- X. D. Dai, Y. Kong, J. H. Li, and B. X. Liu. Extended finnis-sinclair potential for bcc and fcc metals and alloys. *Journal of Physics: Condensed Matter*, 18:4527–4542, 2006.
- M. S. Daw and M. I. Baskes. Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. *Phys. Rev. B*, 29:6443–6453, 1984.
- G Dehm. Miniaturized single-crystalline fcc metals deformed in tension: New insights in size-dependent plasticity. *Progress in Materials Science*, 54(6):664–688, August 2009. ISSN 00796425. doi: 10.1016/j.pmatsci.2009.03.005.

- PM Derlet, A Hasnaoui, and H Van Swygenhoven. Atomistic simulations as guidance to experiments. *Scripta Materialia*, 49(7):629–635, 2003.
- M Dobson, RS Elliott, M Luskin, and EB Tadmor. A multilattice quasicontinuum for phase transforming materials: Cascading cauchy born kinematics. *Journal of Computer-Aided Materials Design*, 14(1):219–237, 2007.
- M. Dobson, M. Luskin, and C. Ortner. Accuracy of quasicontinuum approximations near instabilities. *J. Mech. Phys. Solids*, 58:1741–1757, 2010a.
- M. Dobson, M. Luskin, and C. Ortner. Sharp stability estimates for the force-based quasicontinuum approximation of homogeneous tensile deformation. *Multiscale Model. Simul.*, 8:782–802, 2010b.
- M. Dobson, M. Luskin, and C. Ortner. Stability, instability, and error of the force-based quasicontinuum approximation. *Arch. Ration. Mech. Anal.*, 197:179–202, 2010c.
- L. M. Dupuy, E. B. Tadmor, R. E. Miller, and R. Phillips. Finite-temperature quasicontinuum: Molecular dynamics without all the atoms. *Physical Review Letters*, 95:060202, 2005.
- B. Eidel and A. Stukowski. A variational formulation of the quasicontinuum method based on energy sampling in clusters. *J. Mech. Phys. Solids*, 57:87–108, 2009.
- J. Ericksen. The cauchy and born hypotheses for crystals. In M. Gurtin, editor, *Phase Transformations and Material Instabilities in Solids*, pages 61–77. Academic Press, 1984.
- M. I. Espanol, D. M. Kochmann, S. Conti, and M. Ortiz. A gamma-convergence analysis of the quasicontinuum method. *SIAM Multiscale Modeling and Simulation*, 11:766–794, 2013.
- C. Fang and X. Yang. Study of nanocontact and incipient nanoscratch process using the quasicontinuum method. *Materials Science and Engineering: A*, 600(0):221 – 230, 2014. ISSN 0921-5093. doi: <http://dx.doi.org/10.1016/j.msea.2014.02.027>. URL <http://www.sciencedirect.com/science/article/pii/S0921509314001671>.
- F. Feyel and J.-L. Chaboche. Fe2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre sic/ti composite materials. *Computer Methods in Applied Mechanics and Engineering*, 183(3-4):309 – 330, 2000. ISSN 0045-7825. doi: 10.1016/S0045-7825(99)00224-8. URL <http://www.sciencedirect.com/science/article/pii/S0045782599002248>.
- S. P. A. Gill. The effect of surface-stress on the concentration of stress at nanoscale surface flaws. *International Journal of Solids and Structures*, 44(2223):7500 – 7509, 2007. ISSN 0020-7683. doi: <http://dx.doi.org/10.1016/j.ijsolstr.2007.04.018>. URL <http://www.sciencedirect.com/science/article/pii/S0020768307002004>.

- J R Greer, Warren C Oliver, and W D Nix. Size dependence of mechanical properties of gold at the micron scale in the absence of strain gradients. *Acta Materialia*, 53(6):1821–1830, April 2005.
- J R Greer, D Jang, J.-Y. Kim, and M J Burek. Emergence of New Mechanical Functionality in Materials via Size Reduction. *Advanced Functional Materials*, 19(18):2880–2886, September 2009. ISSN 1616301X. doi: 10.1002/adfm.200900854. URL <http://doi.wiley.com/10.1002/adfm.200900854>.
- Julia R. Greer and Jeff Th.M. De Hosson, Jeff Th.M. Plasticity in small-sized metallic systems: Intrinsic versus extrinsic size effect. *Progress in Materials Science*, 56(6):654–724, August 2011. ISSN 00796425. doi: 10.1016/j.pmatsci.2011.01.005.
- Julia R. Greer and William D Nix. Nanoscale gold pillars strengthened through dislocation starvation. *Physical Review B*, 73(24):1–6, June 2006.
- MikhailA. Grekov and AnnaA. Yazovskaya. Surface stress in an elastic plane with a nearly circular hole. In Holm Altenbach and Nikita F Morozov, editors, *Surface Effects in Solid Mechanics*, volume 30 of *Advanced Structured Materials*, pages 81–94. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-35782-4. doi: 10.1007/978-3-642-35783-1_7. URL http://dx.doi.org/10.1007/978-3-642-35783-1_7.
- M. Gunzburger and Y. Zhang. A quadrature-rule type approximation to the quasi-continuum method. *Multiscale Model. Simul.*, 8(2):571–590, 2010.
- S. Hai and E. B. Tadmor. Deformation twinning at aluminum crack tips. *Acta Materialia*, 51: 117131, 2003.
- Susan Hert and Michael Seel. dD convex hulls and delaunay triangulations. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.6 edition, 2015. URL <http://doc.cgal.org/4.6/Manual/packages.html#PkgConvexHullDSummary>.
- J.E. Inglesfield. Theory of surface structure and bonding. volume 4 of *Cohesion and Structure*, pages 63 – 119. North-Holland, 1995. doi: [http://dx.doi.org/10.1016/S0922-7725\(06\)80003-0](http://dx.doi.org/10.1016/S0922-7725(06)80003-0). URL <http://www.sciencedirect.com/science/article/pii/S0922772506800030>.
- M. Iyer and V. Gavini. A field theoretical approach to the quasi-continuum method. *Journal of the Mechanics and Physics of Solids*, 59(8):1506 – 1535, 2011. ISSN 0022-5096. doi: 10.1016/j.jmps.2010.12.002. URL <http://www.sciencedirect.com/science/article/pii/S0022509610002425>.
- Sergei Izvekov and Gregory A. Voth. A multiscale coarse-graining method for biomolecular systems. *The Journal of Physical Chemistry B*, 109(7):2469–2473, 2005. doi: 10.1021/jp044629q. URL <http://pubs.acs.org/doi/abs/10.1021/jp044629q>.

- Dongchan Jang, Lucas R Meza, Frank Greer, and Julia R Greer. Fabrication and deformation of three-dimensional hollow ceramic nanostructures. *Nature materials*, 12(10):893–898, 2013.
- A.T. Jennings, M.J. Burek, and J.R. Greer. Microstructure versus size: Mechanical properties of electroplated single crystalline cu nanopillars. *Physical Review Letters*, page 135503, 2010.
- J. Jin, S.A. Shevlin, and Z.X. Guo. Multiscale simulation of onset plasticity during nanoindentation of al (001) surface. *Acta Materialia*, 56(16):4358 – 4368, 2008. ISSN 1359-6454. doi: <http://dx.doi.org/10.1016/j.actamat.2008.04.064>. URL <http://www.sciencedirect.com/science/article/pii/S1359645408003418>.
- R. A. Johnson. Alloy models with the embedded-atom method. *Phys. Rev. B*, 39:12554–12559, Jun 1989. doi: 10.1103/PhysRevB.39.12554. URL <http://link.aps.org/doi/10.1103/PhysRevB.39.12554>.
- Mark T. Jones and Paul E. Plassmann. Adaptive refinement of unstructured finite-element meshes.
- C. L. Kelchner, S. J. Plimpton, and J. C. Hamilton. Dislocation nucleation and defect structure during surface indentation. *Phys. Rev. B*, 58:11085–11088, 1998.
- D Kiener, C. Motz, T. Schoberl, M. Jenko, and G Dehm. Determination of Mechanical Properties of Copper at the Micron Scale. *Advanced Engineering Materials*, 8(11):1119–1125, November 2006.
- D Kiener, W Grosinger, G Dehm, and R Pippan. A further step towards an understanding of size-dependent crystal plasticity: In situ tension experiments of miniaturized single-crystal copper samples. *Acta Materialia*, 56(3):580–592, February 2008.
- D Kiener, Christian Motz, G Dehm, and Reinhard Pippan. Overview on established and novel FIB based miniaturized mechanical testing using in-situ SEM. *International Journal of Materials Research (formerly Zeitschrift fuer Metallkunde)*, 100(08):1074–1087, 2009.
- J.-Y. Kim.-Y. Kim, D. Jang, and J. R. Greer. Strength dependence on orientation, size, and deformation path in molybdenum nano-pillars. *International Journal of Plasticity*, 2011.
- W. K. Kim, M. Luskin, D. Perez, A. F. Voter, and E. B. Tadmor. Hyper-qc: An accelerated finite-temperature quasicontinuum method using hyperdynamics. *J. Mech. Phys. Solids*, 63:94–112, 2014a. doi: <http://dx.doi.org/10.1016/j.jmps.2013.10.001>.
- Woo Kyun Kim, M Luskin, D Perez, AF Voter, and EB Tadmor. Hyper-qc: An accelerated finite-temperature quasicontinuum method using hyperdynamics. *Journal of the Mechanics and Physics of Solids*, 63:94–112, 2014b.
- C. Kittel. *Introduction to Solid State Physics*. Wiley, New York, 1996.

- J. Knap and M. Ortiz. An analysis of the quasicontinuum method. *J. Mech. Phys. Solids*, 49(9): 1899–1923, September 2001.
- J. Knap and M. Ortiz. Effect of indenter-radius size on au (001) nanoindentation. *Phys. Rev. Lett.*, 90:226102, Jun 2003. doi: 10.1103/PhysRevLett.90.226102. URL <http://link.aps.org/doi/10.1103/PhysRevLett.90.226102>.
- D.M. Kochmann and G. Venturini. A meshless quasicontinuum method based on local maximum-entropy interpolation. *Model. Simul. Mater. Sci. Eng.*, 22:034007, 2014.
- Y. Kulkarni, J. Knap, and M. Ortiz. A variational approach to coarse graining of equilibrium and non-equilibrium atomistic description at finite temperature. *Journal of the Mechanics and Physics of Solids*, 56:1417–1449, 2008.
- S. Kwon, Y. Lee, J. Y. Park, D. Sohn, J. H. Lim, and S. Im. An efficient three-dimensional adaptive quasicontinuum method using variable-node elements. *Journal of Computational Physics*, 228(13):4789–4810, 2009. doi: 10.1016/j.jcp.2009.03.028. URL <http://dx.doi.org/10.1016/j.jcp.2009.03.028>.
- W. K. Liu, H. S. Park, D. Q., E. G. Karpov, H. Kadowaki, and G. J. Wagner. Bridging scale methods for nanomechanics and materials. *Computer Methods in Applied Mechanics and Engineering*, 195(13-16):1407–1421, 2006. doi: 10.1016/j.cma.2005.05.042. URL <http://www.sciencedirect.com/science/article/pii/S0045782505002884>.
- M. Luskin and C. Ortner. An analysis of node-based cluster summation rules in the quasicontinuum method. *SIAM J. Numer. Anal.*, 47(4):3070–3086, 2009.
- R. Maass, S. Van Petegem, D. Grolimund, H. Van Swygenhoven, D. Kiener, and G. Dehm. Crystal rotation in cu single crystal micropillars: In situ laue and electron backscatter diffraction. *Applied Physics Letters*, 92, 2008.
- J. Marian, J. Knap, and G.H. Campbell. A quasicontinuum study of nanovoid collapse under uniaxial loading in ta. *Acta Materialia*, 56(10):2389 – 2399, 2008. ISSN 1359-6454. doi: <http://dx.doi.org/10.1016/j.actamat.2008.01.050>. URL <http://www.sciencedirect.com/science/article/pii/S1359645408000657>.
- J Marian, G Venturini, B L Hansen, J Knap, M Ortiz, and G H Campbell. Finite-temperature extension of the quasicontinuum method using langevin dynamics: entropy losses and analysis of errors. *Modelling and Simulation in Materials Science and Engineering*, 18(1):015003, 2010.
- Jaime Marian, Jaroslaw Knap, and Michael Ortiz. Nanovoid deformation in aluminum under simple shear. *Acta Materialia*, 53(10):2893 – 2900, 2005. ISSN 1359-6454. doi: <http://dx.doi.org/>

- 10.1016/j.actamat.2005.02.046. URL <http://www.sciencedirect.com/science/article/pii/S1359645405001497>.
- J. Marshall and K. Dayal. Atomistic-to-continuum multiscale modeling with long-range electrostatic interactions in ionic solids. *J. Mech. Phys. Solids*, 2013. in press.
- A. V. Masket. Solid angle contour integrals, series, and tables. *Review of Scientific Instruments*, 28(3):191–197, 1957. doi: <http://dx.doi.org/10.1063/1.1746479>. URL <http://scitation.aip.org/content/aip/journal/rsi/28/3/10.1063/1.1746479>.
- M.I. Mendelev and A.H. King. The interactions of self-interstitials with twin boundaries. *Philosophical Magazine*, 93(10-12):1268–1278, 2013. doi: [10.1080/14786435.2012.747012](http://dx.doi.org/10.1080/14786435.2012.747012). URL <http://dx.doi.org/10.1080/14786435.2012.747012>.
- C. Miehe, J. Schröder, and J. Schotte. Computational homogenization analysis in finite plasticity simulation of texture development in polycrystalline materials. *Computer Methods in Applied Mechanics and Engineering*, 171:387 – 418, 1999.
- R. Miller, M. Ortiz, R. Phillips, V. Shenoy, and E.B. Tadmor. Quasicontinuum models of fracture and plasticity. *Engineering Fracture Mechanics*, 61:427–444, 1998.
- R. E. Miller and V. B. Shenoy. Size-dependent elastic properties of nanosized structural elements. *Nanotechnology*, 11(3):139, 2000. URL <http://stacks.iop.org/0957-4484/11/i=3/a=301>.
- R. E. Miller and E. B. Tadmor. A unified framework and performance benchmark of fourteen multiscale atomistic/continuum coupling methods. *Modelling Simul. Mater. Sci. Eng.*, 17:053001, 2009.
- R. E. Miller and E. B. Tadmor. Qc reference manual, 2011.
- Ronald Miller and E.B. Tadmor. The quasicontinuum method: Overview, applications and current directions. *Journal of Computer-Aided Materials Design*, 9:203–239, 2002. ISSN 0928-1045. URL <http://dx.doi.org/10.1023/A:1026098010127>. 10.1023/A:1026098010127.
- Amir K. Miri, Reza Avazmohammadi, and Fuqian Yang. Effect of surface stress on the deformation of an elastic half-plane containing a nano-cylindrical hole under a surface loading. *Journal of Computational and Theoretical Nanoscience*, 8(2):231–236, 2011. doi: [doi:10.1166/jctn.2011.1683](http://dx.doi.org/10.1166/jctn.2011.1683). URL <http://www.ingentaconnect.com/content/asp/jctn/2011/00000008/00000002/art00012>.
- Y. Mishin, M. J. Mehl, D. A. Papaconstantopoulos, A. F. Voter, and J. D. Kress. Structural stability and lattice defects in copper: Ab initio, tight-binding, and embedded-atom calculations. *Phys. Rev. B*, 63:224106, May 2001. doi: [10.1103/PhysRevB.63.224106](http://dx.doi.org/10.1103/PhysRevB.63.224106). URL <http://link.aps.org/doi/10.1103/PhysRevB.63.224106>.

- L. C. Montemayor, L. R. Meza, and J. R. Greer. Design and fabrication of hollow rigid nanolattices via two-photon lithography. *Advanced Engineering Materials*, 16(2):184–189, 2014. ISSN 1527-2648. doi: 10.1002/adem.201300254. URL <http://dx.doi.org/10.1002/adem.201300254>.
- A.K. Nair, D.H. Warner, R.G. Hennig, and W.A. Curtin. Coupling quantum and continuum scales to predict crack tip dislocation nucleation. *Scripta Materialia*, 63(12):1212 – 1215, 2010. ISSN 1359-6462. doi: 10.1016/j.scriptamat.2010.08.038. URL <http://www.sciencedirect.com/science/article/pii/S1359646210005750>.
- D Norfleet, D Dimiduk, S Polasik, M Uchic, and M Mills. Dislocation structures and their relationship to strength in deformed nickel microcrystals. *Acta Materialia*, 56(13):2988–3001, August 2008.
- C. Ortner and A. V. Shapeev. Analysis of an energy-based atomistic/continuum approximation of a vacancy in the 2D triangular lattice. *Math. Comp.*, 82:2191–2236, 2013. ISSN 0025-5718. doi: 10.1090/S0025-5718-2013-02687-7.
- C. Ortner and L. Zhang. Atomistic/continuum blending with ghost force correction. published online, 2014.
- Christoph Ortner. A priori and a posteriori analysis of the quasinonlocal quasicontinuum method in 1d. *Math. Comp.*, 80(275):1265–1285, 2011. ISSN 0025-5718. doi: 10.1090/S0025-5718-2010-02453-6. URL <http://dx.doi.org/10.1090/S0025-5718-2010-02453-6>.
- H. S. Park, E. G. Karpov, W. K. Liu, and P. A. Klein. The bridging scale for two-dimensional atomistic/continuum coupling. *Philosophical Magazine*, 85(1):79–113, 2005. doi: doi:10.1080/14786430412331300163. URL <http://www.ingentaconnect.com/content/tandf/tphm/2005/00000085/00000001/art00005>.
- J. Y. Park and S. Im. Adaptive nonlocal quasicontinuum for deformations of curved crystalline structures. *Phys. Rev. B*, 77:184109, 2008. doi: 10.1103/PhysRevB.77.184109. URL <http://link.aps.org/doi/10.1103/PhysRevB.77.184109>.
- T Parthasarathy, Satish I Rao, D Dimiduk, M Uchic, and D Trinkle. Contribution to size effect of yield strength from the stochastics of dislocation source lengths in finite samples. *Scripta Materialia*, 56(4):313–316, February 2007.
- Rob Phillips, David Rodney, Vivek Shenoy, Ellad Tadmor, and Michael Ortiz. Hierarchical models of plasticity: dislocation nucleation and interaction. *Modelling Simul. Mater. Sci. Eng.*, 7:769–780, 1999.

- Philippe Poncharal, Z. L. Wang, Daniel Ugarte, and Walt A. de Heer. Electrostatic deflections and electromechanical resonances of carbon nanotubes. *Science*, 283(5407):1513–1516, 1999. doi: 10.1126/science.283.5407.1513. URL <http://www.sciencemag.org/content/283/5407/1513.abstract>.
- S. Prudhomme, P. T. Bauman, and J. T. Oden. Error control for molecular statics problems. *International Journal for Multiscale Computational Engineering*, 4(5-6):647–662, 2006. ISSN 1543-1649.
- N.M. Pugno and E.G. Aifantis. A note on the transition from nano- to mega-mechanics: The role of the stress quantization. *Journal of the Mechanical Behavior of Materials*, 19:1–104, 2011. doi: 10.1515/JMBM.2009.19.1.31.
- Satish I Rao, D Dimiduk, M. Tang, T. A. Parthasarathy, M. D. Uchic, and C. Woodward. Estimating the strength of single-ended dislocation sources in micron-sized single crystals. *Philosophical Magazine*, 87(30):4777–4794, October 2007.
- Satish I Rao, D Dimiduk, T.A. Parthasarathy, M.D. Uchic, M. Tang, and C. Woodward. Athermal mechanisms of size-dependent crystal flow gleaned from three-dimensional discrete dislocation simulations. *Acta Materialia*, 56(13):3245–3259, 2008.
- Maria-Cecilia Rivara. Lepp-bisection algorithms, applications and mathematical properties. *Appl. Numer. Math.*, 59(9):2218–2235, September 2009. ISSN 0168-9274. doi: 10.1016/j.apnum.2008.12.011. URL <http://dx.doi.org/10.1016/j.apnum.2008.12.011>.
- M.C. Rivara, D Pizarro, and N Chrisochoides. Parallel refinement of tetrahedral meshes using terminal-edge bisection algorithms. In *13th International Meshing Roundtable*, 2004.
- P.J. Rous. Surface crystallography: The experimental data base. volume 4 of *Cohesion and Structure*, pages 1 – 61. North-Holland, 1995. doi: [http://dx.doi.org/10.1016/S0922-7725\(06\)80002-9](http://dx.doi.org/10.1016/S0922-7725(06)80002-9). URL <http://www.sciencedirect.com/science/article/pii/S0922772506800029>.
- R. E. Rudd and J. Q. Broughton. Coarse-grained molecular dynamics: Nonlinear finite elements and finite temperature. *Physical Review B*, 72:144104, 2005a.
- Robert E. Rudd and Jeremy Q. Broughton. Coarse-grained molecular dynamics: Nonlinear finite elements and finite temperature. *Phys. Rev. B*, 72:144104, Oct 2005b. doi: 10.1103/PhysRevB.72.144104. URL <http://link.aps.org/doi/10.1103/PhysRevB.72.144104>.
- T. A. Schaedler, A. J. Jacobsen, A. Torrents, A. E. Sorensen, J. Lian, J. R. Greer, L. Valdevit, and W. B. Carter. Ultralight metallic microlattices. *Science*, 334(6058):962–965, 2011. doi: 10.1126/science.1211649. URL <http://www.sciencemag.org/content/334/6058/962.abstract>.

- J. Schröder. *Homogenisierungsmethoden der nichtlinearen Kontinuumsmechanik unter Beachtung von Stabilitätsproblemen*. PhD thesis, Universität Stuttgart, Germany, 2000. Habilitation thesis.
- Wenzhe Shan and Udo Nackenhorst. An adaptive femd model coupling approach. *Computational Mechanics*, 46(4):577–596, 2010. ISSN 0178-7675. doi: 10.1007/s00466-010-0503-3. URL <http://dx.doi.org/10.1007/s00466-010-0503-3>.
- Z W Shan, Ju Li, Y.Q. Cheng, A M Minor, S A Syed Asif, O L Warren, and Evan Ma. Plastic flow and failure resistance of metallic glass: Insight from in situ compression of nanopillars. *Physical Review B*, 77(15):1–6, April 2008. ISSN 1098-0121.
- V. Shenoy, V. Shenoy, and R. Phillips. Finite temperature quasicontinuum methods. *Materials Research Society Symposium Proceedings*, 538:465471, 1999a.
- V. B. Shenoy, R. Miller, E. B. Tadmor, R. Phillips, and M. Ortiz. Quasicontinuum models of interfacial structure and deformation. *Phys. Rev. Lett.*, 80:742–745, 1998.
- V.B. Shenoy, R. Miller, E.b. Tadmor, D. Rodney, R. Phillips, and M. Ortiz. An adaptive finite element approach to atomic-scale mechanics the quasicontinuum method. *Journal of the Mechanics and Physics of Solids*, 47(3):611 – 642, 1999b. ISSN 0022-5096. doi: [http://dx.doi.org/10.1016/S0022-5096\(98\)00051-9](http://dx.doi.org/10.1016/S0022-5096(98)00051-9). URL <http://www.sciencedirect.com/science/article/pii/S0022509698000519>.
- M. S. Shephard, C. Picu, and D. K. Datta. Composite grid atomistic continuum method: An adaptive approach to bridge continuum with atomistic analysis. *International Journal for Multiscale Computational Engineering*, 2(3), 2004. ISSN 1543-1649.
- Jonathan Richard Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.
- Jonathan Richard Shewchuk. Adaptive Precision Floating-Point Arithmetic and Fast Robust Geometric Predicates. *Discrete & Computational Geometry*, 18(3):305–363, October 1997.
- Jonathan Richard Shewchuk. Two discrete optimization algorithms for the topological improvement of tetrahedral meshes. *unpublished manuscript*, 2002.
- L.E. Shilkrot, Ronald E. Miller, and William A. Curtin. Multiscale plasticity modeling: coupled atomistics and discrete dislocation mechanics. *Journal of the Mechanics and Physics of Solids*, 52(4):755 – 787, 2004.

- Will Shroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit*. Kitware, 2006. ISBN 978-1-930934-19-1.
- Hang Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. on Mathematical Software*, 41(2), Feb 2015. doi: 10.1145/2629697.
- G. Simmons and H. Wang. *Single Crystal Elastic Constants and Calculated Aggregate Properties*. MIT Press, Camebridge, MA, 1971.
- G.S. Smith, E.B. Tadmor, N. Bernstein, and E. Kaxiras. Multiscale simulations of silicon nanoin-dentation. *Acta Materialia*, 49(19):4089 – 4101, 2001. ISSN 1359-6454. doi: [http://dx.doi.org/10.1016/S1359-6454\(01\)00267-1](http://dx.doi.org/10.1016/S1359-6454(01)00267-1). URL <http://www.sciencedirect.com/science/article/pii/S1359645401002671>.
- M. R. Sorensen and A. F. Voter. Temperature-accelerated dynamics for simulation of infrequent events. *The Journal of Chemical Physics*, 112(21):9599–9606, 2000. doi: <http://dx.doi.org/10.1063/1.481576>. URL <http://scitation.aip.org/content/aip/journal/jcp/112/21/10.1063/1.481576>.
- P. Suryanarayana. *Coarse-graining kohn-sham density functional theory*. PhD thesis, California Institute of Technology, 2011.
- Phanish Suryanarayana, Kaushik Bhattacharya, and Michael Ortiz. Coarse-graining kohnsham density functional theory. *Journal of the Mechanics and Physics of Solids*, 61(1):38 – 60, 2013. ISSN 0022-5096. doi: <http://dx.doi.org/10.1016/j.jmps.2012.09.002>. URL <http://www.sciencedirect.com/science/article/pii/S0022509612001949>.
- B. A. Szajewski and W. A. Curtin. Analysis of spurious image forces in atomistic simulations of dislocations. *Modelling and Simulation in Materials Science and Engineering*, 23(2):025008, 2015. URL <http://stacks.iop.org/0965-0393/23/i=2/a=025008>.
- E. Tadmor, F. Legoll, W. Kim, L. Dupuy, and R. Miller. Finite-temperature quasi-continuum. *Applied Mechanics Reviews*, 65(1):010803, 2013. doi: 10.1115/1.4023013. URL <http://hal.archives-ouvertes.fr/hal-00849048>.
- E.B. Tadmor, M. Ortiz, and R. Phillips. Quasicontinuum analysis of defects in solids. *Philos. Mag. A*, 73(6):1529–1563, June 1996.
- Z. Tang, H. Zhao, G. Li, and N. R. Aluru. Finite-temperature quasicontinuum method for multiscale analysis of silicon nanostructures. *Physical Review B*, 74:064110, 2006.
- G. I. Taylor. The mechanism of plastic deformation of crystals. part i. theoretical. *Proceedings of the Royal Society of London A*, 145:362–387, 1934.

- B.D. Todd and R.M. Lynden-Bell. Surface and bulk properties of metals modelled with sutton-chen potentials. *Surface Science*, 281(12):191 – 206, 1993. ISSN 0039-6028. doi: [http://dx.doi.org/10.1016/0039-6028\(93\)90868-K](http://dx.doi.org/10.1016/0039-6028(93)90868-K). URL <http://www.sciencedirect.com/science/article/pii/003960289390868K>.
- G. Venturini. *Topics in Multiscale Modeling of Metals and Metallic Alloys*. PhD thesis, California Institute of Technology, 2010.
- G. Venturini, K. Wang, I. Romero, M.P. Ariza, and M. Ortiz. Atomistic long-term simulation of heat and mass transport. *Journal of the Mechanics and Physics of Solids*, 73(0):242 – 268, 2014. ISSN 0022-5096. doi: <http://dx.doi.org/10.1016/j.jmps.2014.09.008>. URL <http://www.sciencedirect.com/science/article/pii/S002250961400194X>.
- A. F. Voter. Parallel replica method for dynamics of infrequent events. *Phys. Rev. B*, 57:R13985–R13988, Jun 1998. doi: 10.1103/PhysRevB.57.R13985. URL <http://link.aps.org/doi/10.1103/PhysRevB.57.R13985>.
- A. F. Voter, F. Montalenti, and T. C. Germann. Extending the time scale in atomistic simulation of materials. *Annual Review of Materials Research*, 32(1):321–346, 2002. doi: 10.1146/annurev.matsci.32.112601.141541. URL <http://www.annualreviews.org/doi/abs/10.1146/annurev.matsci.32.112601.141541>.
- A.F. Voter. A method for accelerating the molecular dynamics simulation of infrequent events. *The Journal of Chemical Physics*, 106(11):4665–4677, 1997. doi: 10.1063/1.473503. URL <http://link.aip.org/link/?JCP/106/4665/1>.
- Xiangyang Wang and Xu Guo. Quasi-continuum model for the finite deformation of single-layer graphene sheets based on the temperature-related higher order cauchy-born rule. *Journal of Computational and Theoretical Nanoscience*, 10(1):154–164, 2013.
- Y Morris Wang, Frederic Sansoz, Thomas LaGrange, Ryan T Ott, Jaime Marian, Troy W Barbee Jr, and Alex V Hamza. Defective twin boundaries in nanotwinned metals. *Nature materials*, 12(8):697–702, 2013.
- J. B. Whitley, G. L. Kulcinski, H. V. Smith, and P. Wilkes. Effects of bombarding ions on the void swelling profile in nickel. In *Effects of Radiation on Structural Materials*, pages 125–143. ASTM International, 1979.
- E. W. Wong, P. E. Sheehan, and C. M. Lieber. Nanobeam mechanics: Elasticity, strength, and toughness of nanorods and nanotubes. *Science*, 277(5334):1971–1975, 1997. doi: 10.1126/science.277.5334.1971. URL <http://www.sciencemag.org/content/277/5334/1971.abstract>.

- S. Xiao and W. Yang. A temperature-related homogenization technique and its implementation in the meshfree particle method for nanoscale simulations. *International Journal for Numerical Methods in Engineering*, 69(10):2099–2125, 2007. ISSN 1097-0207. doi: 10.1002/nme.1841. URL <http://dx.doi.org/10.1002/nme.1841>.
- Shaoping Xiao and Weixuan Yang. A nanoscale meshfree particle method with the implementation of the quasicontinuum method. *International Journal of Computational Methods*, 2(03):293–313, 2005.
- SP Xiao and Ted Belytschko. Material stability analysis of particle methods. *Advances in Computational Mathematics*, 23(1-2):171–190, 2005.
- Q. Yang, Biyikli. E., and A.C. To. Multiresolution molecular mechanics: Statics. *Computer Methods in Applied Mechanics and Engineering*, 258(0):26 – 38, 2013. ISSN 0045-7825. doi: <http://dx.doi.org/10.1016/j.cma.2013.01.014>. URL <http://www.sciencedirect.com/science/article/pii/S004578251300025X>.
- Q. Yang, E. Biyikli, and A. C. To. Multiresolution molecular mechanics: Convergence and error structure analysis. *Computer Methods in Applied Mechanics and Engineering*, 269(0):20 – 45, 2014. ISSN 0045-7825. doi: <http://dx.doi.org/10.1016/j.cma.2013.10.012>. URL <http://www.sciencedirect.com/science/article/pii/S0045782513002648>.
- Qingcheng Yang and Albert C To. Multiresolution molecular mechanics: A unified and consistent framework for general finite element shape functions. *Computer Methods in Applied Mechanics and Engineering*, 283:384–418, 2015.
- W. Yang and S. Xiao. The applications of meshfree particle methods at the nanoscale. In VaidyS. Sunderam, GeertDick Albada, PeterM.A. Slood, and Jack Dongarra, editors, *Computational Science ICCS 2005*, volume 3516 of *Lecture Notes in Computer Science*, pages 284–291. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-26044-8. doi: 10.1007/11428862_40. URL http://dx.doi.org/10.1007/11428862_40.
- C. K. Yap and T. Dube. The exact computation paradigm. In *Computing in Euclidean Geometry*, pages 452–492. World Scientific, Singapore, 1995.
- Xiaowei Zeng and Shaofan Li. A multiscale cohesive zone model and simulations of fractures. *Computer Methods in Applied Mechanics and Engineering*, 199(912):547 – 556, 2010. ISSN 0045-7825. doi: <http://dx.doi.org/10.1016/j.cma.2009.10.008>. URL <http://www.sciencedirect.com/science/article/pii/S0045782509003533>.
- T. Zhu, J. Li, S. Ogata, and S. Yip. Mechanics of ultra-strength materials. *MRS Bulletin*, 34(3): 167–172, 2009. ISSN 0883-7694.

Yong Zhu, Qingquan Qin, Feng Xu, Fengru Fan, Yong Ding, Tim Zhang, Benjamin J. Wiley, and Zhong Lin Wang. Size effects on elasticity, yielding, and fracture of silver nanowires: *In situ* experiments. *Phys. Rev. B*, 85:045443, Jan 2012. doi: 10.1103/PhysRevB.85.045443. URL <http://link.aps.org/doi/10.1103/PhysRevB.85.045443>.